CA Business Service Insight

Manuel d'implémentation

8.2

La présente documentation, qui inclut des systèmes d'aide et du matériel distribués électroniquement (ci-après nommés "Documentation"), vous est uniquement fournie à titre informatif et peut être à tout moment modifiée ou retirée par CA.

La présente Documentation ne peut être copiée, transférée, reproduite, divulguée, modifiée ou dupliquée, en tout ou partie, sans autorisation préalable et écrite de CA. La présente Documentation est confidentielle et demeure la propriété exclusive de CA. Elle ne peut pas être utilisée ou divulguée, sauf si (i) un autre accord régissant l'utilisation du logiciel CA mentionné dans la Documentation passé entre vous et CA stipule le contraire ; ou (ii) si un autre accord de confidentialité entre vous et CA stipule le contraire.

Nonobstant ce qui précède, si vous êtes titulaire de la licence du ou des produits logiciels décrits dans la Documentation, vous pourrez imprimer ou mettre à disposition un nombre raisonnable de copies de la Documentation relative à ces logiciels pour une utilisation interne par vous-même et par vos employés, à condition que les mentions et légendes de copyright de CA figurent sur chaque copie.

Le droit de réaliser ou de mettre à disposition des copies de la Documentation est limité à la période pendant laquelle la licence applicable du logiciel demeure pleinement effective. Dans l'hypothèse où le contrat de licence prendrait fin, pour quelque raison que ce soit, vous devrez renvoyer à CA les copies effectuées ou certifier par écrit que toutes les copies partielles ou complètes de la Documentation ont été retournées à CA ou qu'elles ont bien été détruites.

DANS LES LIMITES PERMISES PAR LA LOI APPLICABLE, CA FOURNIT LA PRÉSENTE DOCUMENTATION "TELLE QUELLE", SANS AUCUNE GARANTIE, EXPRESSE OU TACITE, NOTAMMENT CONCERNANT LA QUALITÉ MARCHANDE, L'ADÉQUATION À UN USAGE PARTICULIER, OU DE NON-INFRACTION. EN AUCUN CAS, CA NE POURRA ÊTRE TENU POUR RESPONSABLE EN CAS DE PERTE OU DE DOMMAGE, DIRECT OU INDIRECT, SUBI PAR L'UTILISATEUR FINAL OU PAR UN TIERS, ET RÉSULTANT DE L'UTILISATION DE CETTE DOCUMENTATION, NOTAMMENT TOUTE PERTE DE PROFITS OU D'INVESTISSEMENTS, INTERRUPTION D'ACTIVITÉ, PERTE DE DONNÉES OU DE CLIENTS, ET CE MÊME DANS L'HYPOTHÈSE OÙ CA AURAIT ÉTÉ EXPRESSÉMENT INFORMÉ DE LA POSSIBILITÉ DE TELS DOMMAGES OU PERTES.

L'utilisation de tout produit logiciel mentionné dans la Documentation est régie par le contrat de licence applicable, ce dernier n'étant en aucun cas modifié par les termes de la présente.

CA est le fabricant de la présente Documentation.

Le présent Système étant édité par une société américaine, vous êtes tenu de vous conformer aux lois en vigueur du Gouvernement des Etats-Unis et de la République française sur le contrôle des exportations des biens à double usage et aux autres réglementations applicables et ne pouvez pas exporter ou réexporter la documentation en violation de ces lois ou de toute autre réglementation éventuellement applicable au sein de l'Union Européenne.

Copyright © 2012 CA. Tous droits réservés. Tous les noms et marques déposées, dénominations commerciales, ainsi que tous les logos référencés dans le présent document demeurent la propriété de leurs détenteurs respectifs.

Support technique

Pour une assistance technique en ligne et une liste complète des sites, horaires d'ouverture et numéros de téléphone, contactez le support technique à l'adresse http://www.ca.com/worldwide.

Table des matières

Chapitre 1 : Introduction 9

Rôles ... 9
- Gestionnaire de catalogues des services/Gestionnaire de détection de services ... 10
- Gestionnaire de contrats ... 10
- Expert en logique applicative ... 11
- Expert en sources de données ... 12
- Administrateur système ... 13

Processus de la solution ... 14
- Planification ... 15
- Conception ... 16
- Implémentation ... 18
- Installation et déploiement ... 18

Chapitre 2 : Planification et conception 19

Session de rassemblement des exigences requises (Tous les rôles) ... 20
Collecte des données échantillon (expert en sources de données) ... 22
Conception ... 23
- Présentation de la conception (gestionnaire de contrats, expert en logique applicative, expert en sources de données) ... 24
- Modélisation de contrat (gestionnaire de contrats) ... 25
- Sorties d'étape de modélisation de contrat (gestionnaire de contrats, expert en sources de données) ... 48
- Modélisation de données (expert en sources de données, expert en logiques applicatives) ... 50
- Sorties de l'étape de modélisation de données (expert en sources de données et expert en logique applicative) ... 68

Chapitre 3 : Implémentation 71

Implémentation : introduction ... 71
Configuration de la structure (gestionnaire de contrats) ... 74
Configuration de la bibliothèque de modèles (gestionnaire de contrats) ... 75
Création de contrats (gestionnaire de contrats) ... 75
- Créer des contrats à partir d'un service ... 77
- Créer des modèles de niveau de service ... 79
- Cycle de vie du contrat et méthodes de contrôle de version ... 80

Collecte de données (expert en sources de données) 82
Fonctionnalité d'adaptateur 83
Environnement d'adaptateur 84
Fichiers principaux 88
Fichiers de travail 89
Communication de l'adaptateur 95
Paramètres de registre de l'adaptateur 98
Modes d'exécution de l'adaptateur 101
Outils de configuration et de maintenance 103
Configuration des adaptateurs et des conversions 104
Conversions automatiques avec des scripts de conversion 145
Rubriques avancées sur les fonctionnalités des adaptateurs 146
Ecriture de scripts de logique applicative (expert en logique applicative) 154
Flux de travaux de l'écriture de scripts de logique applicative 156
Modules de logique applicative 157
Détails sur la logique applicative 158
Activation de contrats (gestionnaire de contrats) 200
Nouveau calcul complet des métriques du contrat 203
Création de livrables (gestionnaire de contrats) 204
Définition des paramètres de sécurité (administrateur) 204
Création de rapports 205
Configuration des pages de tableau de bord 218
Ajout de profils d'alerte de niveau de service 222

Chapitre 4 : Installation et déploiement 227

Introduction 228
Préparations 230
Installation 232
Importation/exportation entre environnements (expert en sources de données) 234
Intégration - Installation du serveur de messagerie (Administrateur système) 236
Définition des préférences système (administrateur système) 238
Paramètres de sécurité (administrateur système) 239
Spécification des paramètres pour la synchronisation SSA 239
Paramètres de partage global 241

Chapitre 5 : Gestion et détection de services 243

Détection de services 243
Fonctionnement de la gestion et de la détection de services 243

Présentation des catégories de services 244
Feuille de calcul des catégories de services 245
Définition de la visibilité et du périmètre de vos services 248
Définition d'une vue des services 249
Définition des options d'affichage des colonnes 250
Choix d'une action pour un service sélectionné 252
Utilisation des options d'attribut et de métadonnées 254
Gestion de vos services 258
Partage des données de comparaison sur Cloud Commons 260
Ajout de votre service à Cloud Commons 261
Ajout de services dans la page Gestion et détection de services 262
Exportation et importation manuelles de données de service 264

Annexe A : Exemples de domaines de service et de catégories de domaine 269

Annexe B : Exemples d'étude de cas 271

Exemples de modélisation de contrat 271
Etude de cas 1 : Disponibilité du système 272
Etude de cas 2 : Disponibilité du système 2 273
Etude de cas 3 : Temps de réponse du système 275
Etude de cas 4 : Performance du centre d'assistance 281
Etude de cas 5 : Sauvegarde de système 283
Exemple de modélisation de métrique financière 284
Etude de cas 6 : Modélisation de conditions financières d'un contrat/service 284
Exemples de modélisation de données 292
Etude de cas 7 : Performance de serveur 292
Etude de cas 8 : Performance du centre d'assistance 296
Exemple d'utilisation d'attributs personnalisés 304
Etude de cas 9 : Cibles multiples dynamiques 304
Exemples de script de conversion 308
Etude de cas 10 : Conversion automatique de base 308
Etude de cas 11 : Mises à jour du modèle de ressource 311
Exemples d'écriture de scripts de logique applicative 314
Etude de cas 12 : Utilisation de la logique de compteur pour calculer le nombre de défaillances 315
Etude de cas 13 : Traitement du groupe de composant dynamique 320
Cas d'étude 14 : Traitement de l'horloge d'accumulation de temps 325
Ecriture d'exemples de logique applicative efficaces 331

Etude de cas 15 : Métriques groupées 334
Etude de cas 16 : Modèles de conception de logique applicative 338
Etude de cas 17 : Fonctionnalité intégrée 343
Etude de cas 18 : Enregistrement 346
Etude de cas 19 : Assistant de configuration d'adaptateur pour une source de données basée sur un fichier 349
Création d'adaptateur 351
Déploiement d'adaptateur 363
Planification d'adaptateur 364
Etude de cas 21 : Exemple d'intégration de LDAP 367
Etude de cas 22 : Génération de rapports à l'aide de PERL 374

Annexe C : Spécifications de la configuration d'adaptateur 377

Structure globale 377
Section General 378
Section d'interface de CA Business Service Insight 385
Section DataSourceInterface 388
Section d'interface de SQL 392
Section InputFormatCollection 397
Section TranslationTableCollection 403
Section TranslatorCollection 405

Annexe D : Définition de formules de logique applicative (Expert en logiques applicatives) 409

Choses à éviter lors de la création de formules de logique applicative 409
Métriques groupées et efficience des ressources 410

Glossaire 413

Chapitre 1 : Introduction

Ce manuel présente le processus d'implémentation et les rôles, responsabilités et interactions impliqués dans la planification, la conception, l'implémentation, ainsi que l'installation et le déploiement de CA Business Service Insight.

Ce chapitre traite des sujets suivants :

Rôles (page 9)
Processus de la solution (page 14)

Rôles

Dans ce document, les rôles décrivent les personnes qui effectuent un ensemble commun de tâches pendant une implémentation standard. Le mot "rôle" peut également désigne l'ensemble des tâches. CA Business Service Insight fournit un nombre prédéfini de rôles que l'administrateur système peut modifier. Vous pouvez également créer d'autres rôles ainsi que des autorisations spécifiques.

Les rôles suivants sont requis lors du processus d'implémentation :

- Gestionnaire de catalogues des services/Gestionnaire de détection de services
- Gestionnaire de contrats
- Expert en logique applicative
- Expert en sources de données
- Administrateur système

Les responsabilités et qualifications de chaque rôle sont décrites ci-après.

Remarque : La même personne peut effectuer plusieurs rôles, comme définis par l'administrateur système.

Gestionnaire de catalogues des services/Gestionnaire de détection de services

Responsabilités :

- Gestion du catalogue de services organisationnels
- Définition du catalogue d'offre de services organisationnels
- Détection et classement des services à l'aide de fonctionnalités de détection de services
- Gestion des exigences pour la définition des contrats et la génération de rapports

Qualifications requises :

- Connaissance du processus de prestation de services E2E de l'organisation
- Connaissance des concepts CA Business Service Insight

Gestionnaire de contrats

Responsabilités :

- Interaction avec l'utilisateur final
- Responsable de la création du catalogue des services d'après les exigences définies
- Création de contrats (avec conservation des contrats existants)
- (Facultatif mais recommandé) Responsable des l'implémentation des accords sur les niveaux de service propres aux clients

Qualifications :

- Compréhension des principes et de la logique des métriques
- Utilisateur avancé des fonctionnalités CA Business Service Insight de l'interface utilisateur graphique

Expert en logique applicative

Responsabilités :

- Implémentation de métriques
- Génération de scripts de logique applicative et conservation des scripts de logique applicative existants

Qualifications :

- Connaissances basiques en développement et bonne connaissance pratique des langages de script (tels que VB-Script)
- Bonne compréhension des flux de données CA Business Service Insight
- Expert en génération de scripts de logique applicative CA Business Service Insight
- Bonne compréhension de l'architecture et des composants CA Business Service Insight
- Utilisateur avancé des fonctionnalités CA Business Service Insight de l'interface utilisateur graphique

Expert en sources de données

Responsabilités :

- Conception d'infrastructures CA Business Service Insight
- Création d'adaptateurs, conservation des adaptateurs existants, interface avec l'infrastructure opérationnelle pour l'obtention de dimensions
- Création et maintien des infrastructures CA Business Service Insight

Qualifications :

- Compréhension de la structure et de l'environnement des sources de données des clients
- Bonnes connaissances pratiques des bases de données, du langage XML et des langages de script
- Compréhension des flux de données CA Business Service Insight et des processus de collecte des données
- Expert en adaptateurs CA Business Service Insight
- Compréhension de l'architecture et des composants CA Business Service Insight
- Utilisateur avancé des fonctionnalités CA Business Service Insight de l'interface utilisateur graphique
- Connaissances en développement préférables

Administrateur système

Responsabilités :

- Gestion de l'installation et des mises à niveau
- Exécution des configurations système (matérielles et logicielles) requises
- Gestion des paramètres de sécurité : définition des utilisateurs et des rôles
- Surveillance et gestion du système

Qualifications :

- Compréhension des réseaux et des infrastructures des clients
- Compréhension de l'architecture et des composants CA Business Service Insight
- Connaissances pratiques des bases de données, du langage XML et des langages de script
- Compréhension des flux de données CA Business Service Insight
- Utilisateur des fonctionnalités CA Business Service Insight de l'interface utilisateur graphique

Processus de la solution

Le processus de la solution inclut trois étapes de base :

- Planification et conception
- Implémentation
- Installation et déploiement

Les rôles décrits précédemment participent à différentes étapes du processus d'implémentation, en fonction de leurs spécialités respectives. Ces rôles interagissent pour assurer la totalité du processus de A à Z, de la définition contractuelle au rapport de sortie.

Ce manuel a été structuré en se basant sur les rôles et les processus, d'après un flux d'étapes générales qui forme le processus d'implémentation. Il décrit la façon dont chaque rôle est impliqué et l'interaction entre chacun d'entre eux.

Les paragraphes suivants présentent les étapes incluses dans le processus d'implémentation et les activités de chaque rôle.

La suite du manuel est structurée par étape avec une indication des rôles participant à chacune d'entre elles. Chaque chapitre indique pour chaque activité le rôle qui en est responsable.

La description suivante présente brièvement les tâches de base de chaque phase et les fonctions des différents rôles. Vous trouverez des détails supplémentaires dans chaque chapitre.

Planification

L'objectif de l'étape de planification est d'identifier et de quantifier tous les aspects qui doivent être résolus dans le cadre de l'implémentation.

Au cours de cette étape, le gestionnaire de contrats recense les besoins de l'entreprise auprès du gestionnaire de catalogues des services pour comprendre les résultats pouvant être attendus suite à une implémentation réussie de CA Business Service Insight. Il est important à ce stade de comprendre les exigences actuelles requises ainsi que les plans futurs pour s'assurer que l'implémentation permettra facilement d'évoluer et de croître en toute transparence.

En fonction des exigences d'implémentation requises, le gestionnaire de contrats doit examiner le processus associé existant (le cas échéant) en vérifiant les entrées et sorties existantes du processus. Il est nécessaire à cette étape d'identifier toutes les sources d'informations requises et d'obtenir des exemples, des contrats, des rapports de sortie et des sources d'entrée utilisés pour calculer les sorties existantes. Ces informations doivent être minutieusement spécifiées pour que le gestionnaire de contrats identifie toutes les entrées nécessaires, afin que la sortie attendue puisse en être dérivée.

Au cours de cette étape, le gestionnaire de contrats examine les contrats afin de s'assurer que le catalogue des services et les modèles fournis dans le cadre de l'implémentation prennent en charge les besoins existants et futurs, et que l'implémentation actuelle couvre tous les domaines du contrat.

Le gestionnaire de contrats examine les rapports et leur format pour s'assurer que toutes les mesures définies sont effectuées afin de pouvoir les produire avec l'implémentation existante.

L'expert en sources de données examine alors les échantillons de données d'entrée par mesures requises et les calculs fournis par le gestionnaire de contrats. L'expert en sources de données peut ainsi identifier tout problème de saisie devant être résolu, comme les formats personnalisés ou les insuffisances, et de déterminer si des sources de données supplémentaires sont nécessaires.

Le but de cet examen est de s'assurer que les sources contiennent les informations requises nécessaires pour calculer les métriques requises et les informations initiales sur le processus de récupération, telles que les méthodes de communication avec les sources de données et la structure des données source.

Conception

Au cours de la phase de conception, les exigences d'entrée et de sortie recueillies sont adaptées à la structure du modèle CA Business Service Insight, qui comprend les contrats, les métriques et les ressources. Cela signifie que les données réelles sont transformées et conceptualisées afin de s'adapter à la structure CA Business Service Insight.

La conception du système comprend les éléments suivants :

Modélisation de contrat

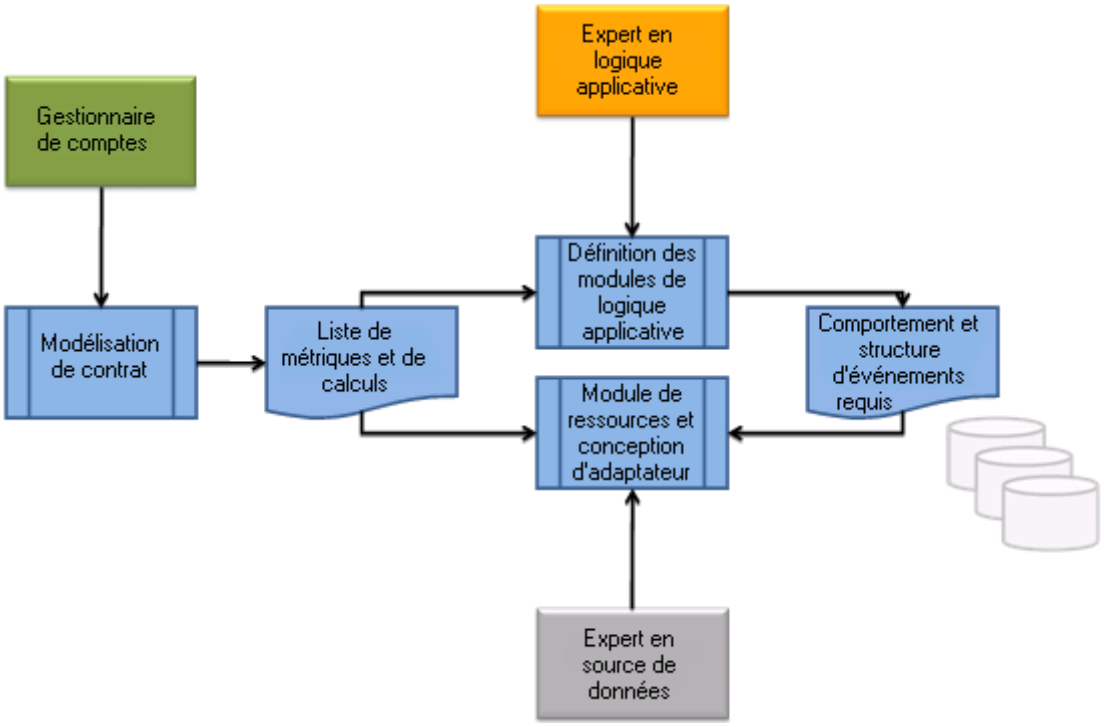
Les accords sur les niveaux de service du client sont interprétés dans des contrats de CA Business Service Insight et des entités de catalogues de services (comme les dossiers de modèles) sont définies. Cette étape est principalement effectuée par le gestionnaire de contrats.

Modélisation de données

Les données de ressource sont examinées et modélisées dans le modèle de ressource CA Business Service Insight. Cette étape est principalement effectuée par l'expert en sources de données et l'expert en logiques applicatives.

Il existe plusieurs méthodes disponibles pour la modélisation des contrats ou des données. Toutefois, il existe bien souvent une pratique optimale qui permet non seulement de résoudre le problème, mais accroît également la flexibilité ou la productivité, en fournissant ainsi une structure robuste pour toute croissance ultérieure.

Pour aider l'expert en sources de données à choisir les méthodes les plus adaptées, des études de cas sont utilisées avec des suggestions de solution.



Comme indiqué dans le diagramme de flux de travaux précédent, comme résultat du processus de modélisation de contrats, le gestionnaire de contrats fournit en sortie la liste Métriques devant être configurée dans le système et leur définition de calcul.

Exemple :

client A, temps de réponse moyen du système CNP.

La liste Métriques est fournie comme entrée pour l'expert en logiques applicatives qui définit la structure et le comportement requis pour les événements, permettant de définir le script de calcul nécessaire.

La liste Métriques et les critères de structure et de comportement d'événement constituent la source d'entrée pour la conception du modèle de ressources et des adaptateurs de données par l'expert en sources de données.

Implémentation

Au cours de la phase d'implémentation, le système CA Business Service Insight est configuré en fonction des résultats de la phase de conception. La phase d'implémentation implique d'utiliser les résultats de l'étape de conception théorique pour créer les critères opérationnels requis pour CA Business Service Insight.

Parmi les éléments à créer ou à résoudre se trouvent notamment :

Gestionnaire de contrats

- Détection de service et installation
- Création de contrats
- Configuration des rapports et des tableaux de bord

Expert en sources de données :

- Configuration de l'adaptateur
- Configuration de l'infrastructure

Expert en logique applicative

- Modules de logique applicative

Installation et déploiement

La phase d'installation et de déploiement concerne l'installation et l'intégration du système de production, les tests et le suivi de ses performances, et la formation des utilisateurs. L'administrateur système et l'expert en sources de données effectuent la plupart des activités à cette étape.

Chapitre 2 : Planification et conception

Ce chapitre traite des sujets suivants :

Session de rassemblement des exigences requises (Tous les rôles) (page 20)
Collecte des données échantillon (expert en sources de données) (page 22)
Conception (page 23)

Session de rassemblement des exigences requises (Tous les rôles)

La session de rassemblement des exigences requises est la première étape critique dans l'implémentation de CA Business Service Insight et doit inclure tout le personnel clé impliqué dans l'implémentation.

Les informations exigées pendant cette session doivent être fournies par des personnes ayant une bonne connaissance des SLA sélectionnés, notamment de la logique applicative des SLA, de leur gestion actuelle, des rapports générés pour ces SLA, et des attentes ultérieures exhaustives en matière de génération de rapports.

Le personnel suivant doit participer :

- Gestionnaires de catalogues des services
- Gestionnaires de contrats
- Expert en sources de données
- Expert en logiques applicatives/génération de scripts
- Tout autre membre du personnel ayant une bonne connaissance des SLA sélectionnés
- Chef de projets (si désigné)

CA recommande les conditions suivantes :

- Il est important de s'assurer que le personnel adéquat est présent et à même de prendre des décisions et clarifier certaines définitions pouvant se présenter lors de l'examen des SLA sélectionnés.
- Il est souhaitable que l'équipe d'implémentation reçoive, avant cette réunion, une copie des SLA choisis pour ce projet. Les participants doivent être familiarisés avec toutes les sources de données pertinentes qui concernent les SLA sélectionnés, pouvoir fournir des images mémoires des données correspondantes et expliquer leur structure intérieure, le cas échéant.

Les objectifs principaux de la session de planification sont la définition et la planification des éléments suivants :

- Portée du travail planifié
- Critères de réussite
- Rôles et responsabilités de toutes les parties impliquées

- Processus d'implémentation et planification
- Configuration matérielle/logicielle requise

Ceci est accompli par :

- Examen des SLA à utiliser pour l'implémentation
- Examen de la "logique applicative" pour chaque objectif des SLA sélectionnés
- Examen de la configuration requise pour les rapports, alertes et tableaux de bord associés aux SLA sélectionnés
- Examen des sources de données pertinentes à utiliser dans l'implémentation
- Examen de la topologie d'infrastructure pertinente
- Collecte des images mémoire de données pertinentes dans les sources de données correspondantes
- Identification des intervenants clés et de leurs responsabilités
- Définition des circuits de communication pour la durée de l'implémentation, réunion de suivi, processus de questions/réponses, et ainsi de suite
- Examen du planning et du processus d'implémentation
- Examen de la situation actuelle, à savoir comment ces SLA sont gérés et quels éléments sont manquants
- Définition des critères de réussite de l'implémentation
- Définition de la configuration matérielle/logicielle requise
- Définition du calendrier d'implémentation

Collecte des données échantillon (expert en sources de données)

Pour effectuer la configuration initiale des adaptateurs hors site, il est important à des fins de collecte de données d'obtenir des échantillons de données passées. La structure de ces échantillons doit être identique à celle des données qui sont ensuite générées par le système réel à partir duquel CA Business Service Insight doit récupérer les données. Outre les échantillons, l'expert en sources de données doit se renseigner sur la source de données auprès de son propriétaire pour être en mesure de clarifier les points suivants :

Type **:**

Base de données, fichier, flux ou autre.

Emplacement **:**

Où est-il ? Comment y accéder (détails de connexion) ? Obstacles à la sécurité ? Plate-forme ?

Convention d'attribution d'un nom **:**

Comment les fichiers sont-ils nommés ? (s'agit-il d'une source de données basée sur un fichier) Comment les tables sont-elles nommées ? Noms des champs de table ?

Disponibilité **:**

Quand peut-on y accéder ? A quel moment doit-on y accéder (considérations de charge) ? Mise à jour continue ou à une fréquence spécifique ? Quelle est la durée de sa persistance ?

Comportement **:**

Les données s'accumulent-elles ? Les données sont-elles écrasées ? Des archives sont-elles conservées ?

Durée **:**

Quelle quantité de données historiques est disponible ?

Volumes **:**

Volume actuel des données ? Taux de croissance ? Prédictions ?

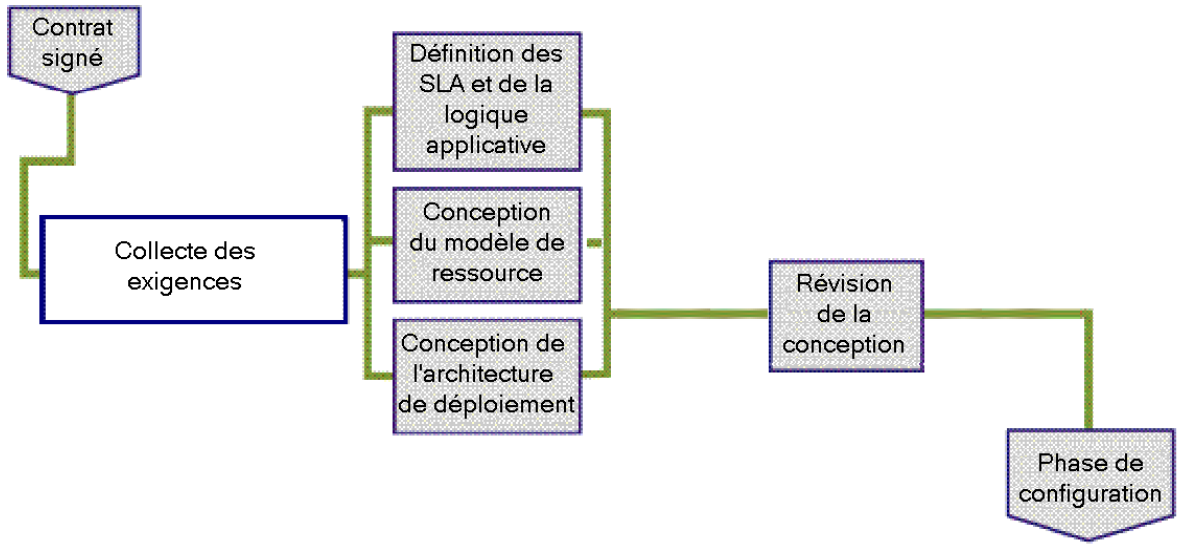
Structure et format **:**

Comment les données sont-elles organisées dans la source de données ? Quels sont les champs de données ? Quels sont les noms de table ? Qu'est-ce qui sépare les champs de données ?

Flux **:**

L'adaptateur obtient-il les données directement ou les données sont-elles d'abord rassemblées par l'adaptateur ?

Conception



Présentation de la conception (gestionnaire de contrats, expert en logique applicative, expert en sources de données)

Cette section explique le processus et le raisonnement derrière la phase de conception du processus de la solution. La phase de conception suit la phase de planification et est elle-même suivie par la phase d'implémentation dans le chapitre suivant.

L'objectif de la phase de conception est que l'équipe d'implémentation puisse effectuer le mappage entre les contrats réels et leurs clauses, et les données de performances existantes dans le système CA Business Service Insight.

Avant le début de ce processus, l'équipe d'implémentation doit connaître les diverses méthodes et options disponibles pour s'assurer que tous les critères requis sont pris en compte, mais également pour s'assurer que la conception obtenue est la plus optimale possible, tout en permettant les évolutions et expansions futures.

Le processus de conception implique que l'équipe d'implémentation effectue les étapes suivantes :

- Examiner les contrats et convertir des contrats en objets CA Business Service Insight correspondants, désignés par le terme "modélisation de contrat" dans le présent manuel ; responsabilité du gestionnaire de contrats.
- A partir des ensembles de données existants, décider des éléments pertinents à extraire et en quelle quantité, en fonction des résultats souhaités, désigné par le terme "modélisation de données" dans le présent manuel ; responsabilité de l'expert en sources de données et de l'expert en logique applicative.

La section Modélisation de contrat explique :

- La terminologie utilisée (essentielle pour une implémentation correcte)
- Les modèles et modules de logique applicative
- Le modèle de niveau de service et son fonctionnement
- L'importance de créer un catalogue des services solide
- Métriques de gestion financière (pénalités, primes et coûts, avec exemples) appliquées aux contrats client et aux autres métriques contractuelles

La section Modélisation de données explique :

- Les événements tels qu'ils s'appliquent à CA Business Service Insight et leur flux à travers le système CA Business Service Insight

- Les métriques et leur enregistrement
- Les ressources CA Business Service Insight et leur identification
- Des suggestions pour l'automatisation de la collecte et de la définition de ces ressources

Tous les points ci-dessus sont expliqués plus en détail dans les parties suivantes.

Il est important d'être conscient que si de mauvais choix sont effectués à cette étape, cela peut avoir des conséquences défavorables sur le fonctionnement de CA Business Service Insight qui peuvent être ensuite difficiles à corriger, voire irréversibles.

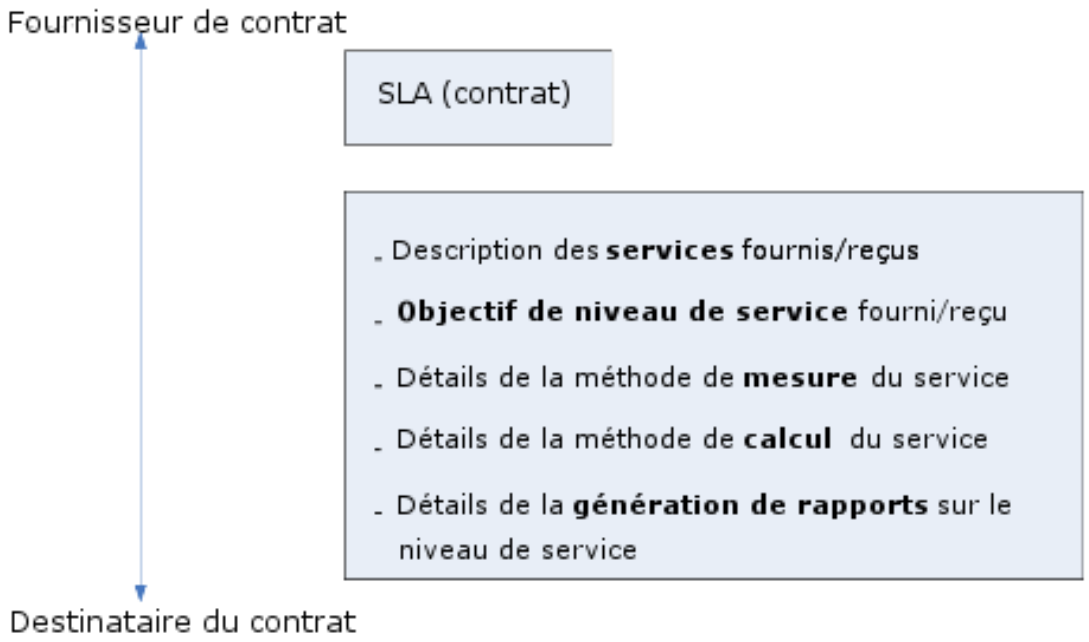
Modélisation de contrat (gestionnaire de contrats)

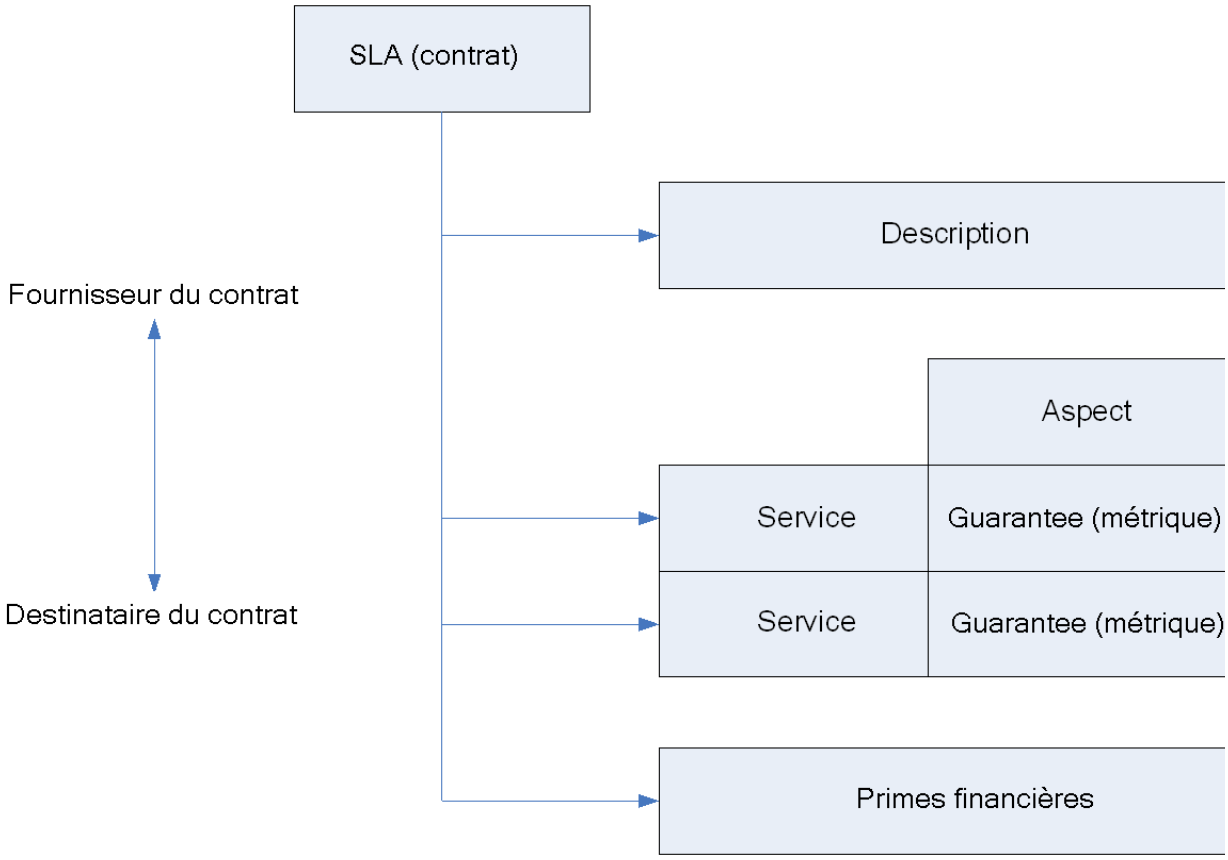
Les tâches suivantes relatives à la modélisation de contrat sont assurées par le gestionnaire de contrats.

Terminologie des contrats

Un contrat est un ensemble d'objectifs. Un objectif est de nature contractuelle ou opérationnelle (métrique) ou de nature financière (prime). Le processus de modélisation de contrat implique de prendre la totalité du contenu du contrat se trouvant dans la portée de l'implémentation et de le transformer en modèle CA Business Service Insight.

Les diagrammes suivants décrivent ce processus.

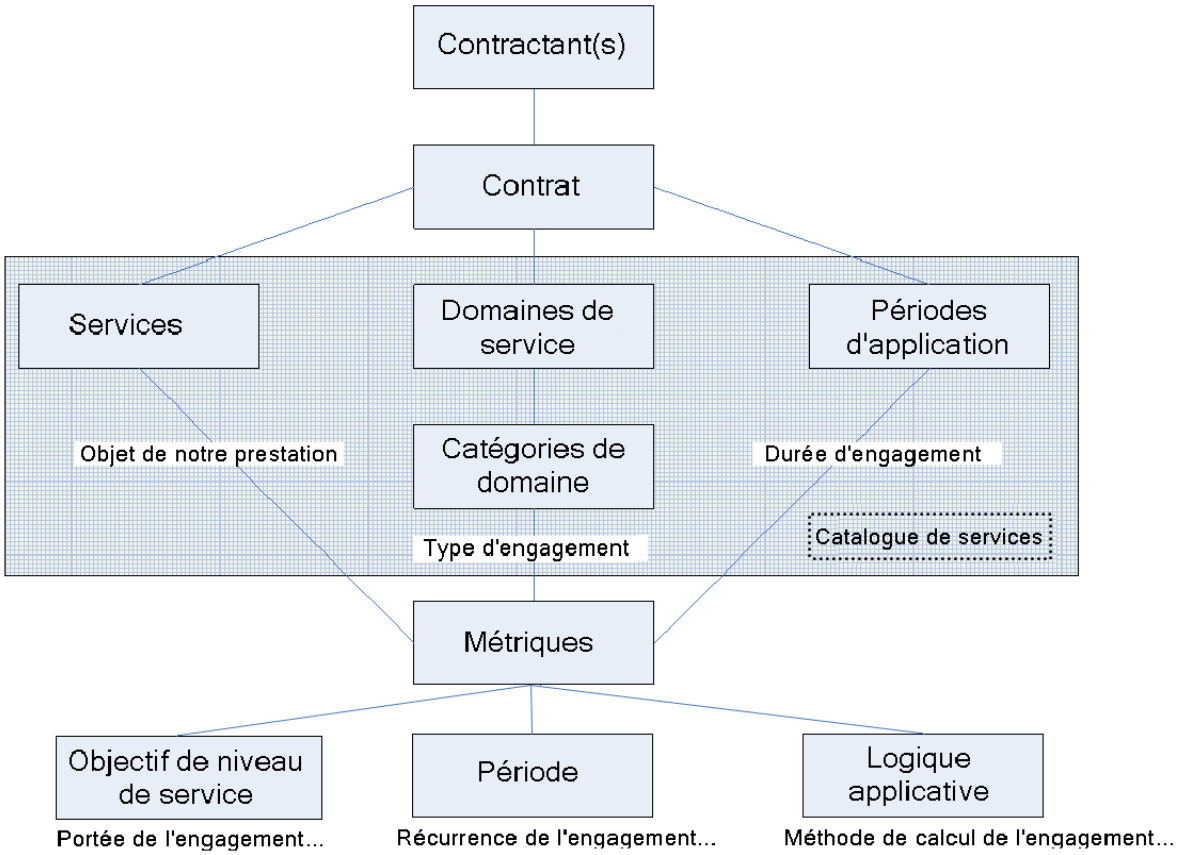




Remarque : Il faut tenir compte des éventuels plans futurs pour le système, même si seuls les aspects se trouvant dans la portée de l'implémentation seront modélisés. Ce cadre doit inclure les critères de gestion généraux requis pour l'organisation pour permettre de concevoir un modèle assez large pouvant inclure les développements futurs. Cela permet de réduire et de simplifier toute modification devant être apportée à l'avenir.

L'adaptation de contrats client au modèle CA Business Service Insight est un processus permettant au gestionnaire de contrats de regrouper les garanties (métriques) proposées dans des limites communes en groupes logiques, composants de service communs, aspects de contrat, etc. Ce groupement logique permet une gestion des niveaux de service très souple.

Le modèle de contrat CA Business Service Insight comprend les entités représentées dans l'illustration suivante :



Contrat

Définit l'accord et l'ensemble de métriques. Un contrat peut être de trois types différents selon la relation entre les contractants impliqués. Il peut s'agir d'un SLA (accord sur les niveaux de service entre l'organisation et ses clients), d'un OLA (contrat opérationnel entre divisions de l'organisation) ou un UC (contrat de sous-traitance entre l'organisation et ses fournisseurs, où généralement l'UC s'applique à un service pour lequel l'organisation fournit ses services à un autre client via un SLA).

Contractant(s)

Définit le client des services fournis, et les parties avec lesquelles ils ont signé le contrat. Un même contractant peut être rattaché à plusieurs contrats. Notez que dans le cadre du contrat vous pouvez définir le fournisseur et le destinataire des services prévus par le contrat.

Métriques

Définit un objectif de niveau de service ou une garantie. Chaque métrique représente une demande de mesure qui fournit le résultat de niveau de service réel sur lequel s'effectue la génération des rapports.

Une métrique comprend les éléments suivants :

Type

Une métrique peut être de l'un des huit types suivants :

SLO	Objectif de niveau de service.
Informatif	Objectif de niveau de service sans cible.
KPI	Indicateur clé de performance, utilisé pour mettre en évidence différents concepts du secteur. Dans le domaine des télécommunications, un indicateur clé de performance est un objectif de niveau de service et constitue une obligation de la part du client. Dans le domaine informatique, il constitue une obligation technique.
KQI	Indicateur clé de qualité, métrique globale permettant de mesurer les résultats agrégés de différents indicateurs.
Intermédiaire	Mesure provisoire à utiliser dans d'autres métriques. Vous ne pouvez pas créer de rapports à partir de ces résultats de métriques.
Consommation	Utilisée par les métriques financières pour calculer la consommation de services/ressources. Utilisée conjointement aux métriques d'élément de prix
Prime	Métrique financière, utilisée pour le calcul des primes (terme positif pour désigner une pénalité).

Elément de prix

Métrique financière, utilisée pour calculer les résultats en fonction des consommations. Différents prix par période ou par nombre d'unités.

Période de suivi

Période de génération de rapports contractuelle ou période pendant laquelle le résultat de niveau de service calculé réel est comparé à la cible. Vous pouvez définir la période de suivi dans CA Business Service Insight avec une granularité en heures, jours, mois, trimestre ou année. A des fins d'analyse de la cause première, outre la période de suivi, la métrique est également calculée pour les six autres granularités, mais les résultats pour ces périodes sont indiqués comme résultats opérationnels uniquement.

Remarque : Cette opération est effectuée uniquement si ces granularités de calcul sont spécifiées pour la métrique.

Période d'application

Période de la période de suivi au cours de laquelle la garantie ou obligation spécifique est applicable, notamment les définitions de périodes d'application exceptionnelles, tels que les fenêtres de maintenance prévue, les jours fériés standard etc.

Logique applicative

Formule qui définit comment évaluer les données brutes afin de calculer la valeur de niveau de service réelle pour cet aspect du service, et les hypothèses spécifiques dont il faut tenir compte dans le calcul. Pendant la phase de conception, seules les grandes lignes du calcul sont identifiées. Le calcul est défini de façon plus détaillée lors de la phase de configuration.

Cible

Obligation de niveau de service contractuelle. La cible peut être obligatoire ou non, selon le type de métrique en question. Dans le cas où une cible n'est pas définie, la métrique fournit des résultats à des fins de génération de rapports uniquement, et elle ne permet pas la comparaison avec une obligation ou une garantie contractuelle. Les cibles peuvent être statiques ou dynamiques.

Remarque : Pour approfondir certains concepts présentés jusqu'ici, consultez l'annexe B - Etudes de cas (page 271).

Chaque métrique est associée aux entités à l'échelle du système suivantes contenues dans le dossier de modèles du système :

Composant de service

Décrit ce qu'il doit fournir.

Groupe de composants de service

Ensemble de composants de service. Un composant de service peut appartenir à plusieurs groupes. Un groupe de composants de service est une entité facultative, qui peut fournir davantage de flexibilité pour la génération de rapports.

Domaine de service

Aspect spécifique de composants de service devant être mesurés à des fins de surveillance des niveaux de service (par exemple, Performances ou Disponibilité).

Catégorie de domaine

Unité de mesure spécifique. Elle définit l'unité de mesure par défaut et indique si l'objectif cible est une valeur minimale ou maximale. Il s'agit essentiellement de la dimension spécifique du domaine de service mesuré (c'est-à-dire, pourcentage de temps de disponibilité, nombre d'interruptions, débit moyen, etc.).

Chacune des entités ci-dessus, ainsi que leurs relations à tous les objectifs de niveau de service surveillés dans CA Business Service Insight, doivent être identifiées pendant l'étape de modélisation de contrat.

Fonctionnement du processus de modélisation

Au cours du processus de modélisation, toutes les métriques devant être configurées dans le système doivent être définies ainsi que leurs entités associées, en fonction du contrat établi et des critères requis pour la génération de rapports.

Avant cette étape ou au cours de celle-ci, il est conseillé de définir une convention d'attribution de noms pour les métriques, pour faciliter l'organisation et la navigation au sein du système. Prenez également en compte la description de la métrique pouvant être utilisée dans la section Enoncé d'objectif de la métrique.

Le processus d'analyse de contrat comprend les étapes suivantes :

1. Identification des objectifs contractuels.

   Pour chaque objectif, identifiez :

   - Un nom approprié (en respectant toute convention de noms préalablement adoptée)
   - Cible (facultative)
   - Période de suivi
   - Unité de mesure
   - Composant de service
   - Domaine de service
   - Catégorie de domaine (et unité de mesure)
   - Période d'application (quand : 24h/24, 7j/7, heures ouvrables uniquement ?)
   - Grandes lignes du calcul (variables nécessaires, mode de calcul du niveau de service)

   Remarque : Certains de ces aspects risquent de ne pas être très clairs du premier coup, mais ils pourront être résolus ultérieurement, lorsque vous affinerez le catalogue des services.

2. Identification, à partir du contrat, de tout objectif financier et détermination des éléments suivants pour chaque objectif financier :
   - S'agit-il d'un objectif de pénalité, de prime ou de coût ?
   - Quelles sont les conditions qui déclenchent l'objectif ?
   - Pour quels composants de service l'objectif se produit-il ?
   - Domaine de service
   - Catégorie de domaine (l'unité dans ce cas peut être une devise, un montant de coût ou un pourcentage de paiement, etc.)

Une fois que tous les objectifs ont été notés et définis, il est conseillé de vérifier la liste complète de métriques et d'essayer d'optimiser/de normaliser le catalogue.

Au cours de cette étape, il est important de veiller à ce que chaque composant de service, domaine de service et catégorie de domaine soit défini de façon logique, afin de constituer un catalogue clair et concis de l'ensemble de l'offre proposée par le système. Cela inclut TOUS les contrats et métriques du système. Vous avez ainsi l'assurance de créer un catalogue des services solide permettant toute expansion future du système.

Les domaines de service et les catégories de domaine n'ont pas besoin d'être définis à un niveau de granularité très détaillé. Ils doivent être descriptifs, mais à un niveau plus élevé que la métrique.

Par exemple, si vous prenez les trois métriques suivantes :

- % de rapports d'interruption livrés à temps
- % de rapports d'exception livrés à temps
- % de rapports de gestion livrés à temps

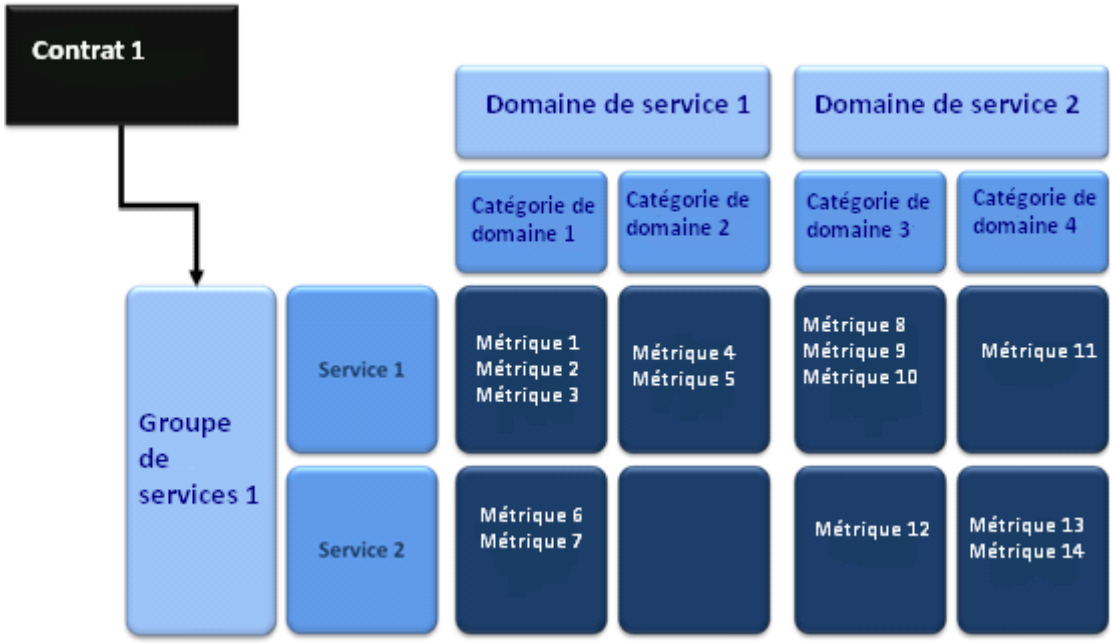
La meilleure catégorie dans laquelle entrent ces trois métriques est probablement le domaine de service Performances et la catégorie de domaine % de rapports livrés à temps. La catégorie de domaine ne doit pas mentionner le type de rapport. Ces trois métriques utilisent probablement la même méthode de calcul et la même logique applicative (à l'exception peut-être d'un paramètre permettant de faire la distinction entre les différents types de rapport).

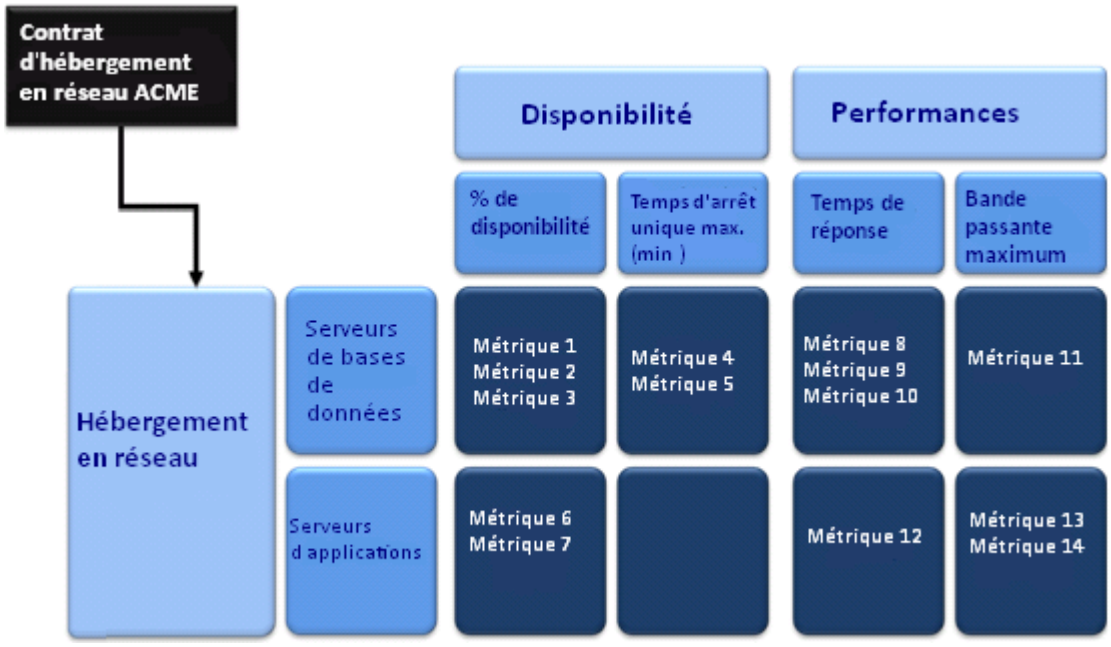
Vous trouverez des domaines de service et des catégories de domaine appropriés dans la bibliothèque de la norme ITIL (Information Technology Infrastructure Library). Les exemples fournis ici le sont uniquement à titre de suggestion, et chaque organisation particulière peut avoir ses propres normes pour ces exemples. Consultez l'annexe A - Exemples de domaines de service et de catégories de domaine (page 269) pour des suggestions de domaines de service et de catégories de domaine.

Remarque : Il peut être utile à ce stade d'organiser une réunion avec tous les intervenants clés, afin d'obtenir confirmation de chacun d'eux que le catalogue choisi prend en charge leurs besoins actuels et futurs.

D'autres exemples qui démontrent des points importants sont présentés dans les annexes. Dans ces exemples, l'un des objectifs de contrat est décrit, ainsi que sa modélisation. Lors de la modélisation de situations réelles, il est nécessaire de prendre en compte tous les objectifs afin que les entités de catalogue les représentent dans leur globalité.

Une fois que le processus d'identification de toutes les métriques et de leurs entités apparentées est terminé, le gestionnaire de contrats dispose d'une matrice qui décrit l'ensemble des métriques, comme illustré dans le schéma ci-dessous.





D'autres questions à prendre en considération dans le processus de modélisation sont décrites dans les parties suivantes.

Questions pour le gestionnaire de contrats

Voici les questions que le gestionnaire de contrats doit prendre en considération afin de s'assurer que tous les aspects ont été pris en compte :

Comment savoir si j'ai choisi le composant de service approprié ?

Si vous pouvez appliquer le composant de service à plus d'un contrat et qu'il peut être mesuré dans plus d'un aspect, cela signifie qu'il a été défini correctement.

Par exemple, le système X est un système fourni à plus d'un client et peut se mesurer par sa disponibilité, sa fiabilité, ses performances, et ainsi de suite.

Comment savoir si j'ai choisi le domaine de service approprié ?

Si vous pouvez mesurer le domaine de service ou le calculer de plusieurs façons, cela signifie qu'il décrit un aspect général du service fourni et qu'il s'agit donc du domaine de service approprié.

Par exemple, vous pouvez mesurer la disponibilité de plusieurs façons, notamment grâce au pourcentage du temps de disponibilité. Elle peut également être mesurée grâce au pourcentage de disponibilité pendant ou en dehors des heures ouvrables, au nombre de défaillances, au temps moyen entre pannes (MTBF), au temps moyen de réparation, au temps d'interruption maximum, au temps d'interruption moyen, au temps d'interruption total, etc. Toutes ces méthodes permettent d'évaluer la disponibilité d'un système particulier.

Cas à prendre en considération lors du processus de modélisation

Voici plusieurs exemples de cas courants ou plus spécifiques décrivant les concepts à prendre en compte dans le processus de modélisation. Ces concepts peuvent permettre d'obtenir une définition plus précise des métriques et une structure stable.

Métriques sans cible

Puisque le domaine cible de la définition de métrique n'est pas obligatoire, lorsqu'elle n'est pas définie, des rapports de niveau de service sont disponibles pour la métrique. Toutefois, aucun rapport de niveau de service par rapport à la cible et à l'écart n'est possible (car il n'y a pas de cible permettant de comparer le niveau de service réel calculé). Ces types de métrique sont définis dans les cas où des rapports sont requis pour des informations qui ne font pas partie des obligations contractuelles réelles.

La définition de ce type de métrique permet à l'utilisateur de disposer de toutes les fonctions d'exploration possibles pour la génération de rapports, et offre au responsable de niveau de service la possibilité d'appliquer les mesures à une cible à tout moment à l'avenir.

***Par exemple* :**

La garantie contractuelle est d'assurer 99 % de disponibilité du réseau et de générer des rapports sur le nombre d'interruptions par mois.

Deux métriques sont définies, l'une avec une cible de 99 % de disponibilité, et l'autre pour compter le nombre d'interruptions pour chaque mois sans cible. Les deux métriques peuvent être intégrées à un rapport, mais seule la première comporte des calculs de déviation à cause de son obligation contractuelle.

Remarque : Une autre méthode possible pour résoudre ce type de situation est d'utiliser des formats de sortie de logique applicative et des rapports libres sur ce type de données. Toutefois, cela ne permet plus la capacité d'exploration du rapport sur les données, ni la possibilité d'utiliser l'assistant de création de rapports simples. L'avantage de l'utilisation de sorties de logiques applicatives est d'autre part d'économiser l'utilisation du moteur, en ayant un nombre de métriques plus restreint.

Pour toutes autres informations sur cette méthode, consultez l'étude de cas Sorties-Tables utilisateur (page 178).

Métriques avec cibles

Dans les cas où une cible est définie pour une métrique, il existe deux façons possibles de spécifier la cible. Vous pouvez la spécifier comme cible statique ou dynamique. Les cibles statiques sont les plus utilisées : il peut s'agir d'une valeur convenue valide pendant toute la durée du contrat.

Par exemple :

La disponibilité de réseau ne doit pas être inférieure à 98 % tous les mois.

La cible dans ce cas est 98 %.

De même, la cible peut dépendre des performances des mois précédents ou juste changer de valeur au cours de l'année. Il existe beaucoup d'autres situations que vous êtes susceptible de rencontrer ici, mais en général toutes sont implémentées avec une formule. CA Business Service Insight prend en charge cette fonctionnalité par un appel à une fonction supplémentaire du modèle de logique applicative standard. La fonction cible peut accéder à d'autres paramètres dans le contexte de la logique applicative et peut prendre en charge tous les scénarios possibles requis.

Par exemple, le temps de résolution des tickets dans le centre d'assistance, qui dépend de la charge du centre d'assistance : le temps moyen de résolution des tickets haute priorité est de 1 jour s'il y a moins de 1 000 tickets au cours du même mois. S'il y a plus de 1 000 tickets envoyés au centre d'assistance dans le mois, le temps de résolution moyen des tickets haute priorité est de 2 jours.

Dans ce cas, la métrique est définie comme ayant une cible dynamique, évaluée dans le cadre du script de logique applicative en fonction du nombre de tickets émis pour ce mois.

Remarque : Pour plus de détails sur la méthode d'implémentation de cibles dynamiques, consultez l'étude de cas Implémentation de cibles dynamiques (page 188).

Paramètre de métrique

Un paramètre de métrique est une valeur pouvant être résolue depuis la logique applicative de la métrique et qui peut facilement être modifiée depuis la définition de la métrique, sans nécessité de modifier le code proprement dit. Il est utilisé à la place de valeurs codées en dur et peut facilement être modifié.

La spécification de paramètres de métrique est primordiale pour l'identification des modules de logique applicative et pour la création de contenu réutilisable. De plus, les paramètres de métriques sont accessibles à travers l'assistant de création de contrats qu'admet qu'un utilisateur final à effectuer change facilement.

***Par exemple* :**

- Les incidents de sévérité 1 doivent être résolus dans les 24 heures.

  Dans l'obligation ci-dessus, la cible est un taux de résolution de 24 heures et vous pouvez définir le niveau de sévérité (Severity1) comme paramètre.

- Le nombre d'interruptions au cours du mois ne doit pas dépasser 3. Est considérée comme une interruption toute période de plus de 5 minutes durant laquelle le système cesse d'être disponible.

  Dans l'obligation ci-dessus, vous pouvez définir comme paramètre la période devant être considérée comme une interruption.

Paramètre de contrat

Un paramètre de contrat est une valeur pouvant être résolue par toutes les métriques d'un contrat. Vous pouvez utiliser un paramètre de contrat dans la métrique à l'aide de la même méthode que pour un paramètre de métrique, mais en le définissant comme paramètre dynamique.

Il est recommandé d'utiliser un paramètre de contrat lorsque plusieurs métriques nécessitent l'utilisation de la même valeur. D'autre part, l'utilisation de paramètres de contrat permet de faciliter la gestion des contrats. Puisque les paramètres ont tendance à changer souvent et exigent d'être actualisés dans le système, il est plus facile d'accéder à un seul point du contrat et de modifier tous les paramètres en même temps que d'accéder à chaque métrique du contrat et de modifier les valeurs des paramètres au niveau de la métrique.

Par conséquent, la modélisation la plus recommandée consiste à définir les paramètres dans le niveau du contrat en tant que paramètres de contrat et d'accéder à leurs valeurs via les paramètres dynamiques au niveau de la métrique.

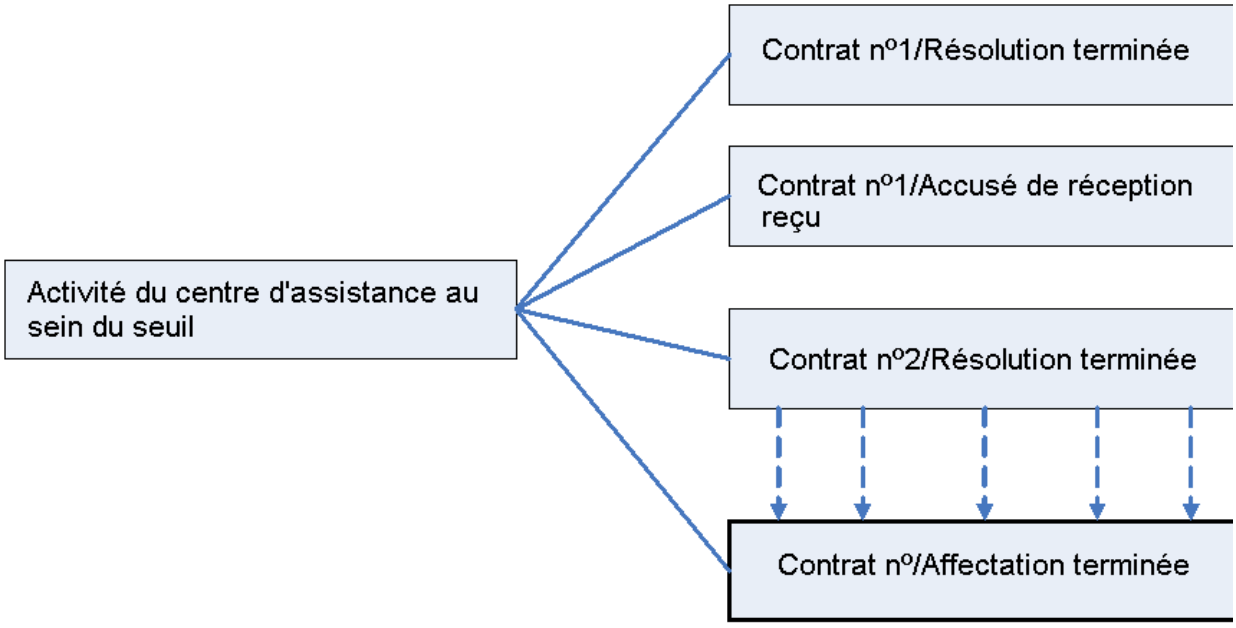
Pour obtenir un exemple, consultez l'étude de cas Performances d'assistance (page 281).

Modèles et modules de logique applicative

Les modèles de logique applicative constituent une façon simple de stocker une méthode de calcul pour une métrique. Ils constituent un composant de logique applicative complet et une façon pratique de créer une base de référence pour d'autres composants de logique applicative. Les nouveaux composants de logique applicative créés depuis un modèle copient le code et créent une nouvelle instance de celui-ci. Cependant, l'utilisation de modèles est peu flexible et il est préférable d'utiliser des modules de logique applicative chaque fois que possible.

Les modules de logique applicative sont des composants indépendants du code qui permettent la réutilisation de la même base de code par d'autres logiques applicatives. Les modules peuvent également inclure d'autres modules ; il peut donc exister plusieurs niveaux de hiérarchie. Lorsque vous utilisez des modules, le code est contenu à un endroit et réutilisé par chacun des autres composants qui y sont reliés. Cette réutilisation des sections de code facilite la maintenance en supprimant la duplication du code et en permettant de modifier rapidement et facilement la logique à l'échelle du système.

Pendant l'étape de conception, il est nécessaire d'identifier les modules de logique applicative principaux et leurs paramètres associés. Une fois que la modélisation de contrats est terminée et que le gestionnaire de contrats a une vision claire de la logique à utiliser, il devient possible d'identifier les calculs qu'ils ont en commun et qui peuvent être définis dans des modules distincts.



Le schéma ci-dessus décrit un module qui calcule le taux de réussite de l'activité du centre d'assistance pour atteindre une cible située dans des seuils donnés. Pour l'implémenter comme décrit, il faut définir deux paramètres, appelés paramètres de métrique : un paramètre qui définit le type d'activité de centre d'assistance et un autre pour le seuil de comparaison (voir la définition d'un *paramètre de métrique* dans la section Cas à prendre en considération lors du processus de modélisation (page 37)).

Par un examen attentif des types de calculs implémentés dans le système, vous remarquerez probablement qu'un certain nombre de types similaires peuvent être effectués en modifiant une petite partie du code, et en utilisant un paramètre permettant d'agir comme "commutateur" entre eux. De cette façon, vous pouvez réduire la quantité de code à recréer et optimiser la quantité de code réutilisé.

Modèles de niveau de service

Un modèle de niveau de service est un ensemble de composants de services regroupés et des métriques associées définies pour les mesurer. Ces modèles de niveau de service peuvent être créés selon les besoins et sont souvent définis en fonction des aspects de service les plus souvent mesurés.

La question clé lorsque vous définissez les modèles de niveau de service est d'identifier et d'exposer tous les paramètres qui pourraient être utilisés pour modifier le comportement des métriques. Cela confère au système la plus grande flexibilité et facilite la configuration pour les utilisateurs du système. Lorsque vous utilisez le modèle de niveau de service pour créer de nouvelles métriques, chacun des paramètres de configuration est présenté à l'interface de l'assistant ; peu de personnalisations (voire aucune) sont nécessaires pour activer le contrat. Les paramètres présentés à l'utilisateur à l'aide de l'assistant se trouvent dans l'énoncé d'objectif. Par conséquent, prenez en compte dans l'énoncé d'objectif de la métrique tous les paramètres que vous souhaitez voir disponibles pour être modifiés par l'utilisateur final.

Pour assurer une efficacité maximale des modèles de niveau de service, essayez d'effectuer toute la logique applicative par le biais de la fonctionnalité de module, comme décrit précédemment.

Voici un exemple de scénario pour l'utilisation des modèles de niveau de service, où des composants de service d'hébergement d'applications sont fournis aux clients dans les trois niveaux de service suivants : Bronze, Silver et Gold, selon le montant que le client est disposé à payer. Des métriques supplémentaires peuvent être fournies dans chaque niveau d'hébergement supérieur, avec des objectifs plus sévères. Chacun de ces niveaux pourrait être un bon candidat pour la création d'un modèle de niveau de service, comme indiqué dans l'exemple de scénario suivant :

- Hébergement d'applications - Bronze (5 métriques)
- Hébergement d'applications - Silver (8 métriques)
- Hébergement d'applications - Gold (12 métriques)

Désormais, lorsqu'un nouveau client est ajouté au système, il est facile de choisir la définition requise en fonction des souhaits du client. L'utilisation de l'interface de l'assistant permet de personnaliser chacune des métriques qui s'y trouvent pour ce client particulier. Notez qu'il est également possible de ne choisir que certaines des métriques à partir des définitions, ou même d'ajouter des métriques issues de deux définitions différentes au nouveau contrat client, si une personnalisation particulière est requise.

Importance de créer un catalogue des services solide

Comme décrit précédemment, le catalogue des services est un composant clé du système et doit être configuré de façon structurée. Cela permet au système d'être utilisé efficacement par tous les utilisateurs et évite la confusion. Il permet également au système de se développer avec l'organisation et de gérer les futures améliorations et ajouts avec un impact minimal sur ce qui est déjà en place.

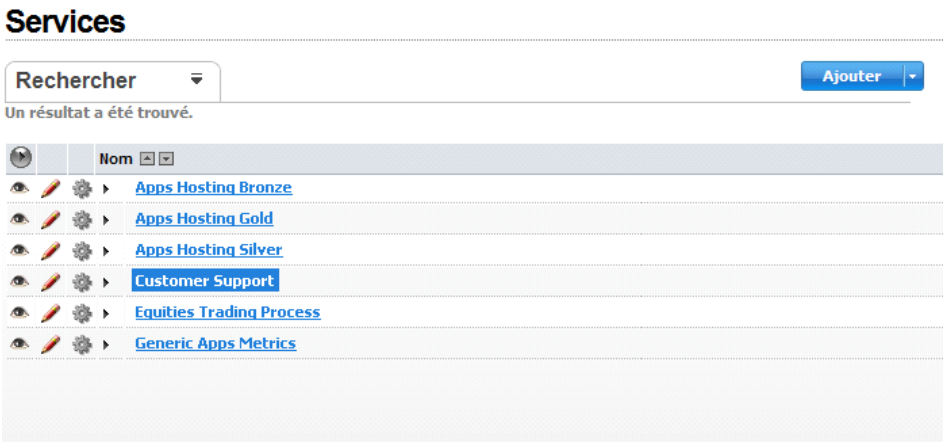
Le système offre un degré élevé de flexibilité pour la création et la gestion du catalogue de composants de services et des SLA. Toutefois, puisque son fonctionnement passe avant tout par une bonne conception, le temps consacré à la planification pour l'avenir est essentiel aux premières étapes de la conception.

La définition du catalogue des services CA Business Service Insight d'une manière efficace et structurée offre à votre système la flexibilité d'ajouter des services et des domaines au catalogue des services ultérieurement. Cela permet également l'ajout ultérieur de contrats, de métriques et de rapports. En outre, elle conduit à une approche plus structurée de la logique applicative et ouvre la voie à la possibilité d'utiliser des modules et des modèles standardisés pour traiter les données commerciales et leurs calculs.

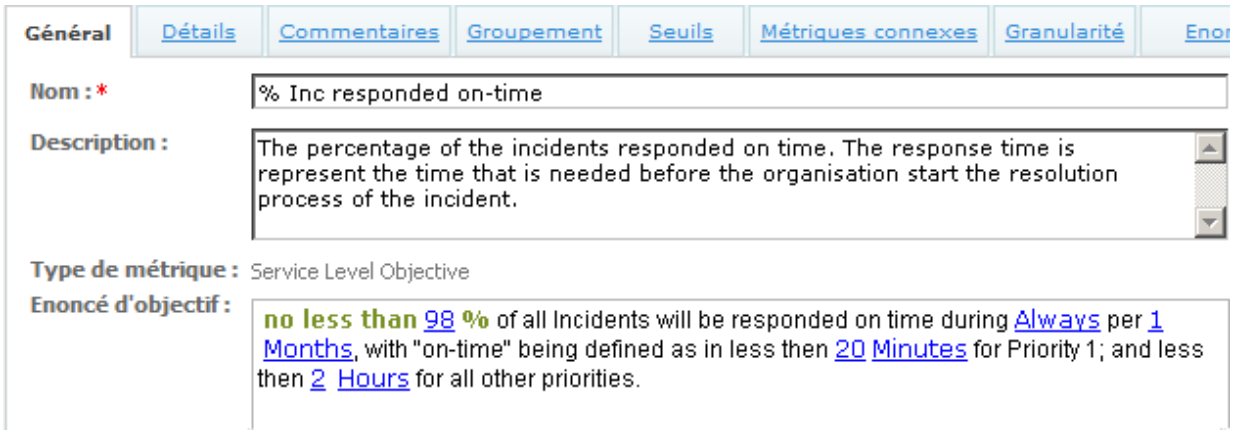
Conjointement au catalogue, les modèles de niveau de service définis dans le système offrent au gestionnaire de contrats un excellent moyen de créer de nouveaux contrats très facilement et avec peu ou aucune connaissance requise des niveaux de données sous-jacentes. Une configuration correcte doit vous permettre de configurer un nouveau contrat pour un client en modifiant ses paramètres dans le modèle de niveau de service uniquement. Tout cela repose, cependant, sur la mise en place la plus efficace possible du catalogue et des définitions. Ces paramètres sont tous exposés via l'énoncé d'objectif pour chaque métrique dans le modèle de niveau de service et peuvent être modifiés à cet endroit, ou à partir de l'assistant lorsque vous utilisez la définition.

Exemple **:**

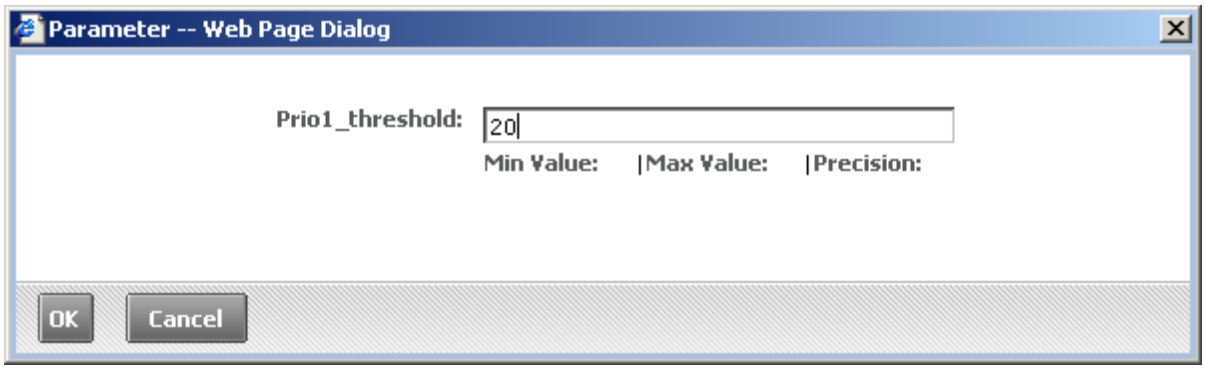
Lorsque vous utilisez un modèle de niveau de service pour déployer des métriques d'assistance client à un contrat, il est possible de sélectionner les métriques requises à partir d'une définition existante



Dans ce modèle de niveau de service, il existe une métrique intitulée '*% des incidents traités à temps*'. Vous voyez que cela comporte un élément de subjectivité, la notion *à temps* pouvant être remise en question. L'exemple suivant explique la mesure effectuée dans cette métrique :



L'énoncé d'objectif affiché au bas de l'onglet Général de la métrique (ou sur l'onglet Enoncé d'objectif) affiche tous les paramètres exposés dans cette métrique. Sur l'illustration précédente, la définition de *à temps* est fixée à 20 minutes. Il s'agit d'un paramètre personnalisable pour permettre notre propre réinterprétation de cette métrique prédéfinie. Pour modifier cette valeur, vous pouvez cliquer sur le lien de paramètre 20 minutes.



De cette façon, vous pouvez personnaliser la nouvelle métrique créée à partir du modèle de niveau de service sans avoir à modifier la logique applicative sous-jacente de la métrique. Notez que cela suppose aussi que la logique applicative soit écrite de manière à intégrer ces paramètres dans le calcul du niveau de service.

Cet exemple simple montre clairement à quel point il est important de créer un ensemble robuste et flexible de modèles de niveau de services pour le catalogue système, afin de permettre leur réutilisation pour les futurs contrats.

Gestion financière (pénalités, primes et coûts)

Les versions antérieures de CA Business Service Insight comportaient des entités de contrat appelées pénalités, qui étaient implémentées à l'aide de formules de type Excel. Les pénalités basaient entièrement leurs résultats sur l'entrée de la métrique du contrat et s'appuyaient sur des fonctions de base pour calculer la quantité de pénalité obtenue. A partir de la version 4.0 et ultérieure, les pénalités ont été remplacées par un ensemble complet de métriques financières que l'utilisateur peut créer, offrant ainsi davantage de flexibilité. Vous pouvez utiliser ces métriques financières pour fournir des informations de prime ou de coût relatives au contrat.

Remarque : La prime remplace l'ancien terme "pénalité" de CA Business Service Insight 3.0 et versions antérieures, et peut être positive ou négative selon les performances. Cependant, une prime négative n'est ni plus ni moins qu'une pénalité. Il est également important de noter que si vous implémentez une pénalité avec le type de métrique Prime, rappelez-vous de faire en *sorte* que la fonction Result() renvoie une valeur négative. Cela permet à toutes les fonctions de synthèse pouvant combiner ensemble différents résultats de métrique, d'ajuster les totaux dans la bonne direction. Cela étant, une prime augmente la valeur tandis qu'une pénalité la diminue.

CA Business Service Insight version 4.0 fournit également la capacité de créer une métrique de consommation mesurant l'utilisation de composants de service et ressources, et également la combiner à une métrique d'élément de prix pour déterminer le coût de ce service ou cette ressource. La combinaison de ces métriques à la fonctionnalité de prévision améliorée rend désormais possible la création de métriques de gestion financière complète.

Les métriques financières peuvent également acquérir leur source d'entrée d'autres métriques contractuelles et déterminer les valeurs de pénalité ou de prime associées en fonction des performances de ces métriques contractuelles. Elles peuvent également utiliser d'autres types d'informations pour déterminer leur résultat, comme le Prix par unité et les modèles de prévision, permettant ainsi des fonctionnalités de génération de rapports de type Coûts prévus par rapport à coûts réels.

***Exemple d'élément de coût* :**

Une application de risque donnée a un coût associé basé sur le nombre d'utilisateurs concurrents du système. Il est calculé mensuellement et une valeur de prévision est fournie pour cette application. Le prix par unité de cette application (coût par utilisateur concurrent) est indiqué dans la table ci-dessous (supposez que cette application entre dans l'index 1) :

Index	Oct-01-2006	Nov-01-2006	Dec-01-2006	Jan-01-2007	Feb-01-2007	Mar-01-2007	Apr-01-2007	May-01-2007	Jun-01-2007
1	1.0	1.0	1.0	0.75	0.75	0.75	0.6	0.6	0.6
2	0.9	0.9	0.9	0.7	0.7	0.7	0.5	0.5	0.5

Le nombre prévu d'utilisateurs concurrents au cours de cette période est également disponible (A nouveau, index 1).

Index	Oct-01-2006	Nov-01-2006	Dec-01-2006	Jan-01-2007	Feb-01-2007	Mar-01-2007	Apr-01-2007	May-01-2007	Jun-01-2007
1	3000	5000	8000	5000	5000	5000	5000	5000	5000
2	2500	2500	6000	4000	4000	4000	4000	4000	4000
3	1000	2000	3000	1000	1000	1000	1000	1000	1000

En modélisant cette métrique de coût à l'aide des tables de coûts, il est possible de déterminer le coût de l'application par mois, en fonction du nombre réel d'utilisateurs concurrents. Cette information provient de la source de données et peut être multipliée par les chiffres du prix par unité ci-dessus pour obtenir les chiffres des coûts. Vous pouvez également la comparer aux valeurs de prévision pour obtenir une analyse du coût prévu par rapport au coût réel. La logique applicative dans ce cas est nécessaire pour déterminer le nombre réel d'utilisateurs concurrents rencontrés au cours de la période et le multiplier par les valeurs de coût par unité. En outre, la fonction de prévision au sein de la logique applicative fait référence aux informations de prévision. Voici un exemple d'application d'une métrique d'élément de coût.

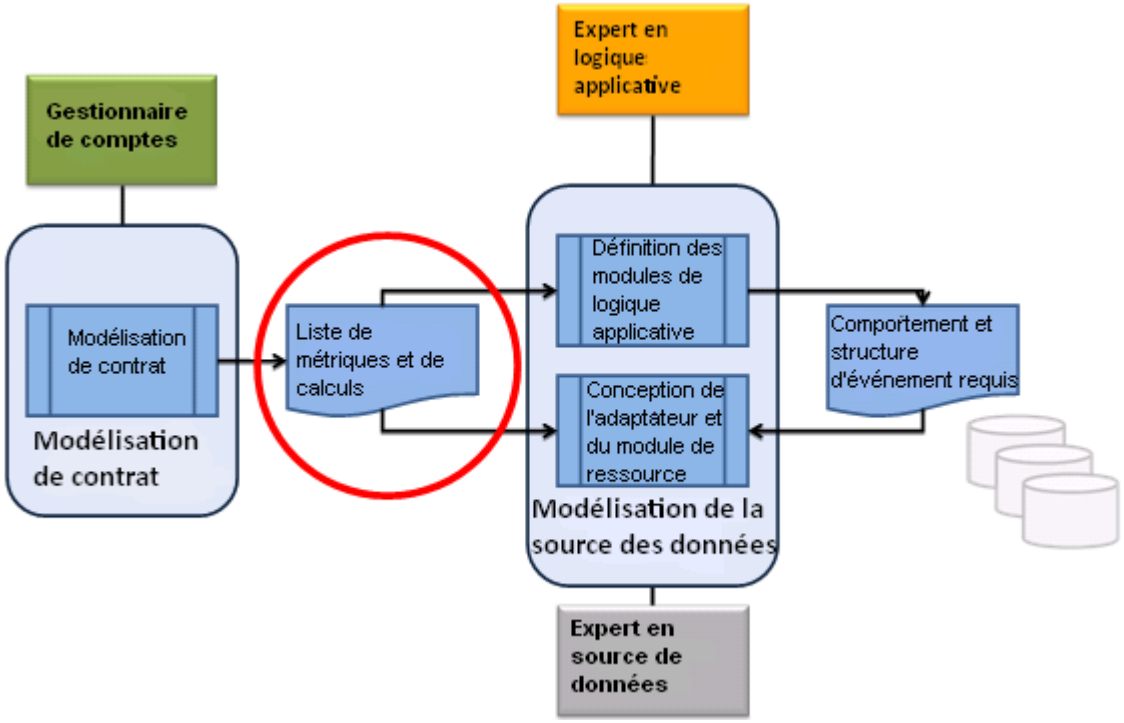
***Exemple de scénario de pénalité* :**

Le SLA client comporte une clause de non-performance incluse permettant de s'assurer que le réseau est disponible 98 % du temps pour un mois donné pendant les heures ouvrables. Un niveau de service mensuel en dessous de ce seuil entraîne le paiement d'une pénalité basée sur une formule (Pénalité = $1000 par chaque pourcentage entier en dessous de la cible (c.-à-d. 96,5 % = (98-Round(96,5)) * 1000 = (98-97) * 1000 = -$1000)).

Pour implémenter cette condition de pénalité, vous pouvez créer une métrique de prime financière issue d'une métrique existante (*Disponibilité du réseau* >= 98 %). Le processus d'enregistrement pour cette métrique utilise le processus RegisterByMetric() pour recevoir les valeurs de niveau de service de cette métrique afin d'effectuer des comparaisons. Cela envoie les valeurs de niveau de service pour la période de suivi de cette métrique dans cette mesure financière sous forme d'événements qui sont ensuite utilisées dans le cadre du calcul pour déterminer le montant de la pénalité pour la même période, à l'aide de la formule du scénario.

Remarque : Pour d'autres études de cas, consultez l'étude de cas de modélisation de métriques financières (page 284).

Sorties d'étape de modélisation de contrat (gestionnaire de contrats, expert en sources de données)



Comme illustré dans le schéma précédent, afin de passer à l'étape de modélisation de données de la solution, le gestionnaire de contrats doit fournir, à la fin de l'étape de modélisation de contrats, la liste des métriques et de leurs calculs requis. La liste inclut les informations suivantes :

- Une définition de catalogue des services complète, comprenant :
  - La liste complète des composants de service proposés
  - La répartition des domaines de services et catégories de domaine
  - La définition de toutes les périodes d'application requises pour chacune des métriques
  - La liste des modules de logique applicative pour gérer chaque type de calcul (notamment les paramètres nécessaires à leur implémentation)
- Une liste complète de métriques à implémenter, notamment :
  - Des conventions d'attribution de noms bien définies pour les métriques
  - La catégorisation de chaque métrique selon les composants de catalogue des services prédéfinis

Ce document est un outil utile pour s'assurer que toutes les définitions clés d'un contrat ont été établies et que toutes ses métriques sont complètement définies.

Les informations de cette feuille de calcul représentent la source de données dont l'expert en sources de données a besoin pour passer à l'étape suivante, la modélisation de données.

Une fois ces éléments terminés, il est possible de passer à l'étape suivante, où vous pouvez commencer la modélisation avec des données réelles issues des sources. Sans modèle de contrat terminé, il est très difficile de savoir exactement ce qui est requis de la source de données pour satisfaire les obligations contractuelles.

Modélisation de données (expert en sources de données, expert en logiques applicatives)

La section Modélisation de données décrit la deuxième partie du processus de conception. La phase de modélisation de données est le processus consistant à prendre les données provenant de sources client, d'identifier des éléments spécifiques des données requises, de décider comment ces données doivent être normalisées et d'évaluer la façon d'attribuer les données pertinentes aux métriques correspondantes (via le processus d'enregistrement).

Cette phase comprend les tâches suivantes :

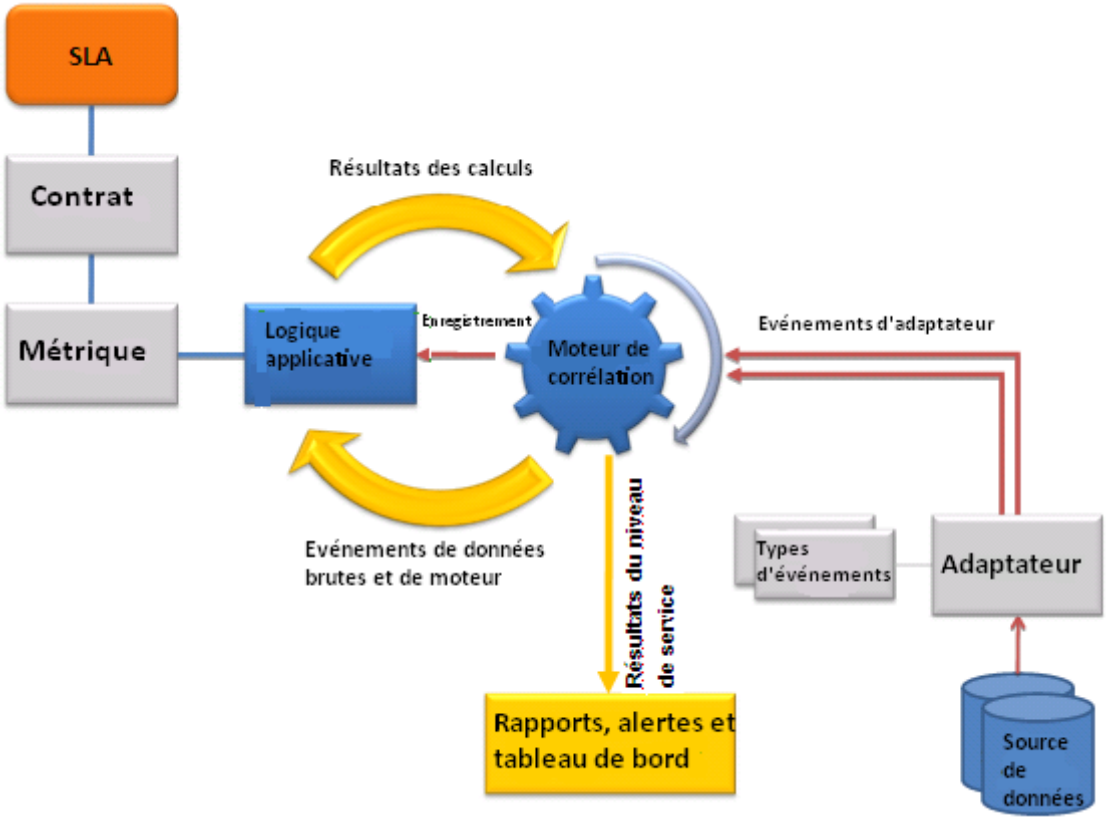
- **Expert en logiques applicatives** :
  - Identification et définition de la structure d'événement d'entrée requise pour le calcul, définie ultérieurement comme types d'événement
  - Création des champs requis dans le type d'événement pour fournir des données requises pour tous les calculs dans lesquels elles sont utilisées
  - Définition du processus d'enregistrement de métrique pour optimiser le flux d'événements
  - Détermination de la façon de créer le modèle de ressources à partir des données disponibles pour satisfaire toutes les exigences de génération de rapports et de logique
- **Expert en sources de données** :
  - Identification du type de données source et décision concernant la façon de les normaliser par l'adaptateur en types d'événement définis et intégrés dans la base de données CA Business Service Insight
  - Choix des identificateurs de type d'événement qui devront être joints aux données

Evénements et flux

Le flux de données dans le système se présente sous forme d'événements. Là où se trouve un événement se trouve un message informatif créé par l'adaptateur en fonction des données sources et dont le format est utilisable par CA Business Service Insight pour ses calculs de niveau de service. Les données brutes sont toujours composées d'événements.

La conception doit par conséquent être axée sur ce flux d'événement dans le système.

Avant modélisation des critères de données, l'expert en logique applicative et l'expert en sources de données doivent avoir une connaissance approfondie des événements et de leur flux dans le système CA Business Service Insight. Le schéma suivant illustre de façon détaillée ce flux d'événement de base.



Le schéma précédent montre comment les événements sont extraits de la source de données par les adaptateurs et normalisés en une structure d'événements standard définie comme le type d'événement. Ces événements sont envoyés par les adaptateurs à CA Business Service Insight. Ils sont désignés sous le terme d'événements de données brutes.

Les calculs de logique applicative dans chaque métrique sont basés sur un sous-ensemble d'événements de données brutes. La logique applicative demande ce sous-ensemble en effectuant l'*enregistrement*.

En fonction de l'instruction d'enregistrement, le moteur de corrélation envoie uniquement les événements de données brutes pertinents pour les calculs de logique applicative.

Les autres types d'événement envoyés à la logique applicative sont les événements de moteur. Tous les concepts impliqués dans ce processus sont expliqués en détail dans ce chapitre.

Cette section se concentre sur les parties suivantes du schéma :

- Source de données
- Adaptateur
- Type(s) d'événement
- Enregistrement de métrique

Le modèle de données CA Business Service Insight a été conçu pour optimiser l'efficacité de ce flux de données dans le système.

En règle générale, CA Business Service Insight fonctionne sur deux couches : la couche d'infrastructure et la couche du modèle applicatif. Pour simplifier la répartition, disons que la couche d'infrastructure comprend les adaptateurs, les ressources et les objets de type d'événement, tandis que la couche applicative comprend les objets de contrats, de métriques et de services. Entre les deux couches se trouve une couche virtuelle, appelée couche de corrélation.

L'objet Type d'événement est un identifiant d'événement. Le type d'événement détermine la façon dont les événements sont définis et signalés à CA Business Service Insight. Cela définit également la structure du champ de données d'événement pour qu'il puisse être interprété par la logique applicative pendant le traitement.

Un autre identificateur d'événement est la ressource qui constitue l'entité la plus petite utilisée dans les calculs. Par exemple, lors du calcul de la disponibilité du serveur, la définition logique de la plus petite entité sur laquelle la génération de rapports est requise peut être un serveur spécifique, ou il peut s'agir d'un client lorsque la génération de rapport s'effectue sur la gestion du ticket envoyé par l'utilisateur. La ressource est une définition d'une entité CA Business Service Insight dérivée aussi bien de la source de données que de la condition de calcul. Un type de ressource est attribué à chaque ressource et constitue un identifiant de ressource qui détermine exactement quel type de ressource est défini. Chaque ressource doit avoir un type de ressources associées, ce qui permet également l'ajout d'attributs personnalisés pouvant être associés à chaque ressource. Pour plus d'informations sur ces attributs, consultez la section Ressources et gestion (page 60).

La corrélation se produit entre les événements d'adaptateurs entrants et les métriques de contrat. La base de ce processus de corrélation est l'allocation de ressources et l'enregistrement de métriques.

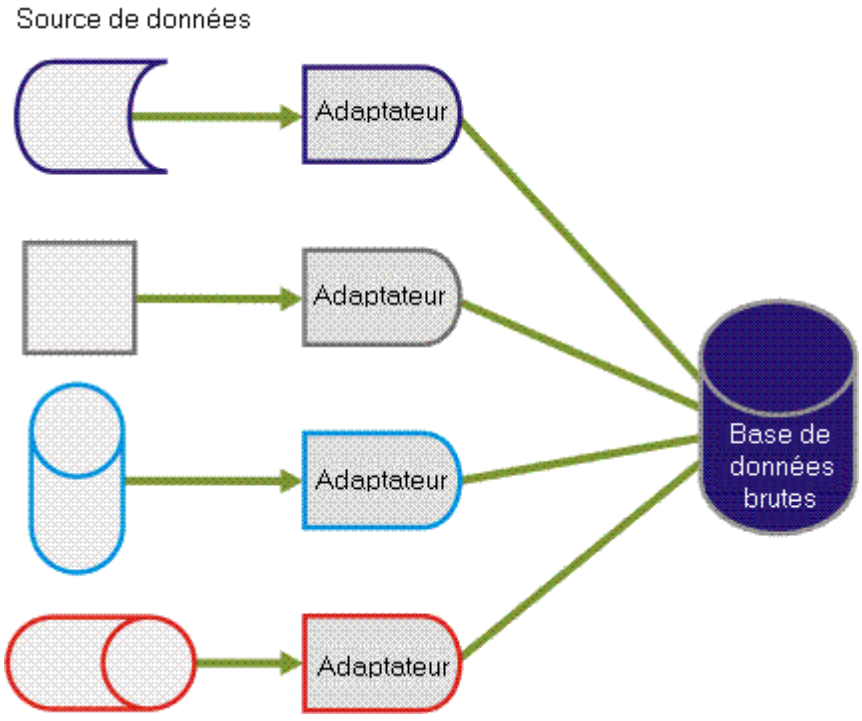
L'allocation de ressources et l'enregistrement de métriques spécifient quels flux d'événement de ressource sont mesurés et par quelle métrique.

Notez ici qu'avec l'enregistrement de métriques, il peut y avoir un degré de réutilisation et de codépendance avec d'autres métriques puisqu'il est possible d'utiliser la sortie d'une métrique comme entrée d'une autre métrique. De même, certains événements intermédiaires ne sont pas utilisés comme sortie d'une métrique pour la mesure de niveau de service, mais plutôt comme une étape intermédiaire de calcul qui peut ensuite être utilisée par d'autres métriques.

Modèle de données - Présentation

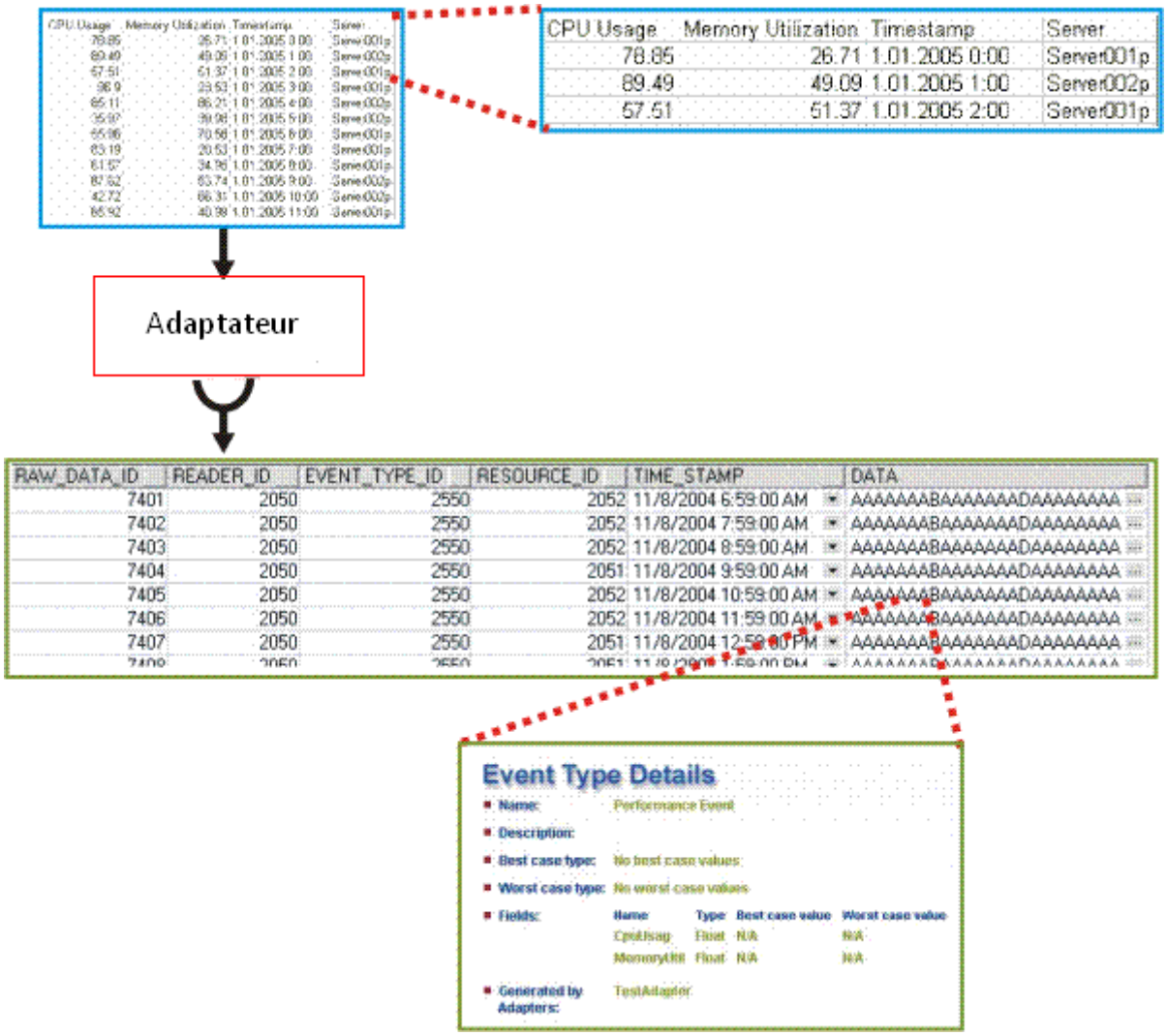
Le modèle de données CA Business Service Insight a été conçu pour relever et surmonter le défi suivant.

Les données brutes sont récupérées par les adaptateurs depuis différentes sources de données disparates et conservées dans de nombreux formats. Ces données diverses doivent être récupérées et harmonisées dans une table de base de données unique. Par conséquent, les adaptateurs doivent lire et normaliser les données dans un modèle de données unifié, comme illustré sur l'illustration suivante.



Dans le cadre de ce processus, tous les champs de données sont insérés dans le même champ de table de base de données, mais ils sont cryptés. Chaque ligne insérée dans la base de données CA Business Service Insight comporte un identifiant de type d'événement auquel elle est rattachée. La définition de type d'événement contient les descriptions des champs de données. Cela permet également au moteur de corrélation d'interpréter correctement les champs de données et d'identifier le moment où ils sont requis par la logique applicative pour les calculs.

L'illustration suivante montre une représentation graphique de la section relative à la récupération des données et à la population de la base de données de ce processus. Y apparaît également une section élargie montrant ce que les données représentent en termes réels, plutôt que la façon dont les données brutes apparaissent.



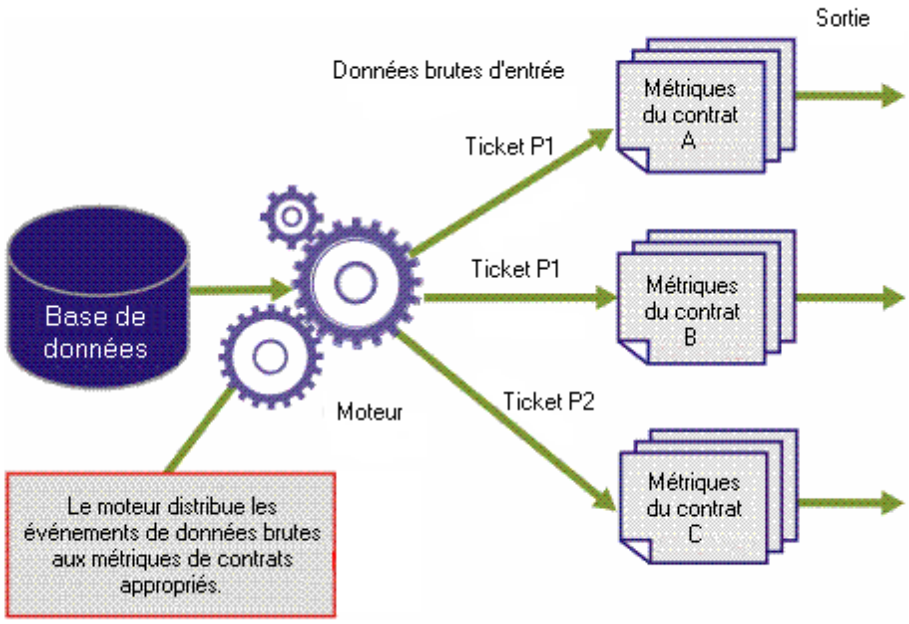
Le système CA Business Service Insight comprend également tous les contrats et métriques nécessitant une évaluation par rapport aux données brutes pour produire les informations de performance de niveau de service résultantes. Chaque métrique doit recevoir uniquement le sous-ensemble de données pertinent pour son calcul. Les données brutes comprennent un nombre potentiellement vaste de documents de divers types. L'utilisation de la métrique pour filtrer les événements pertinents en fonction de leurs valeurs n'est pas très efficace. Par conséquent, le moteur CA Business Service Insight distribue les données brutes pertinentes à chaque métrique spécifique.

Exemple **:**

Pour les deux métriques suivantes d'un contrat :

- Temps de résolution moyen des tickets de priorité 1 (P1)
- Temps de résolution moyen des tickets de priorité 2 (P2)

La première métrique doit évaluer uniquement les tickets de priorité 1 et la deuxième métrique, uniquement ceux de priorité 2. Par conséquent, le moteur doit distribuer les enregistrements en conséquence. De plus, le temps de résolution dans un contrat est calculé pour les tickets P1 ouverts pour le contractant A, alors que dans l'autre contrat, il est calculé pour les tickets P1 pour le contractant B, et dans le troisième contrat, pour les tickets P2 pour le contractant C. Par conséquent, le moteur doit sélectionner le type de ticket et le client pour lequel il a été signalé, comme montré par l'illustration suivante.



Comme expliqué précédemment, les enregistrements de données brutes ont des identifiants associés qui permettent au moteur d'identifier les enregistrements et événements relatifs à chaque métrique de la logique applicative. Les deux identificateurs sont le type d'événement et la ressource.

Procédé de conversion et normalisation

La fonction de l'adaptateur est de lire les données de la source de données et de les normaliser au format d'un événement. Chaque événement dans CA Business Service Insight est composé des champs suivants :

- ID de la ressource
- ID du type d'événement
- Horodatage
- Champs de valeurs en fonction du type d'événement

L'adaptateur doit associer les champs d'origine dans la source de données aux champs de ressource CA Business Service Insight correspondants. Pour cela, il utilise une table de conversion qui contient la valeur trouvée dans la source de données et l'ID de la ressource CA Business Service Insight correspondant.

Le processus d'association de l'ID de la ressource et de l'ID du type d'événement à la valeur de la source de données qui convient est appelé *procédé de conversion*. Au cours de ce processus, la table de conversion est créée avec les valeurs correspondantes. Cette table est utilisée par l'adaptateur afin de renseigner l'ID du type d'événement approprié et les ID de la ressource dans l'enregistrement d'événement qu'il crée. Des tables indépendantes sont créées pour traduire les ressources et les types d'événement.

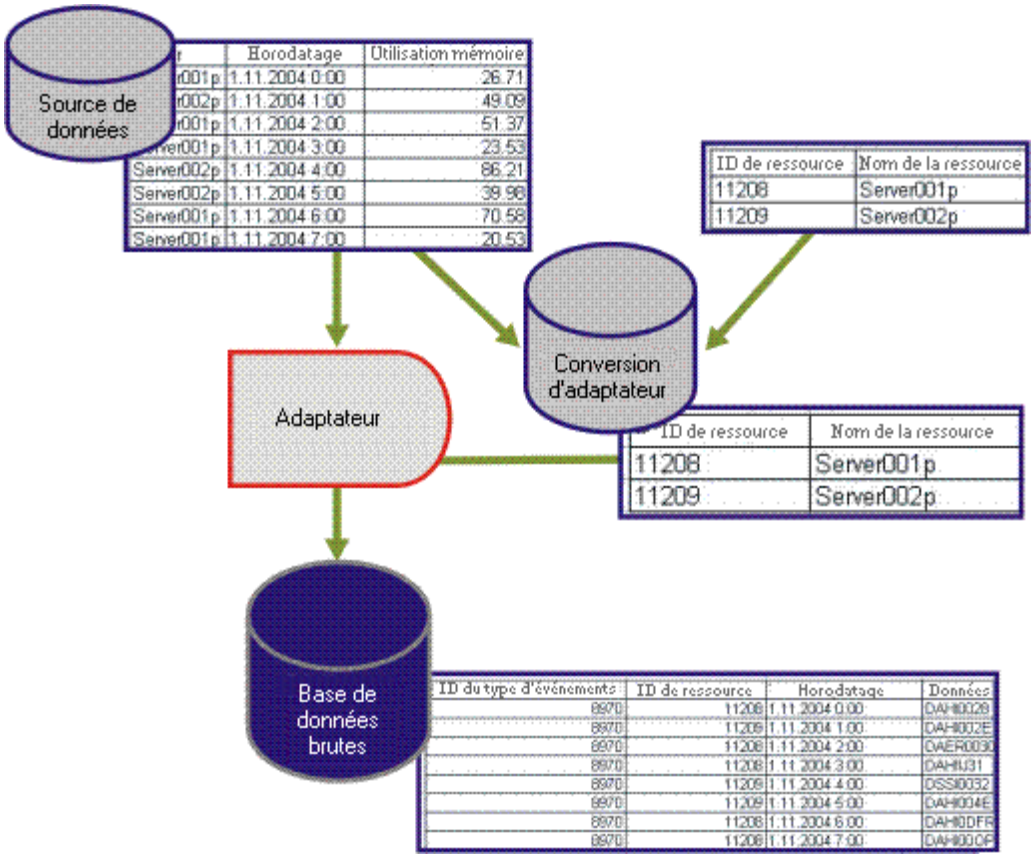
Comme mentionné précédemment, l'ID de la ressource et l'ID du type d'événement sont utilisés à des fins d'enregistrement, tandis que le champ de données et les valeurs d'horodatage sont utilisés pour les calculs réels.

Le champ d'horodatage est également utilisé par le moteur pour déterminer l'ordre et le délai d'exécution des événements envoyés pour calcul.

La définition du type d'événement est effectuée manuellement dans CA Business Service Insight en fonction de la source de données en entrée et de la sortie requise.

Remarque : Vous pouvez réaliser la définition des ressources manuellement ou automatiquement à l'aide de scripts de conversion (consultez la section Ressources et gestion (page 60) pour plus de détails).

L'illustration suivante présente l'interaction entre la source de données, la table de procédé de conversion, l'adaptateur et le tableau de données brutes de CA Business Service Insight.



Enregistrement de métrique

Pour que le moteur de corrélation sache quelles données peuvent être demandées, la métrique doit inscrire sa présence et ses exigences dans le moteur de corrélation.

L'enregistrement de métrique est la demande par une métrique de recevoir des événements, et uniquement ceux qu'elle doit inclure dans son calcul. Cette demande est effectuée par l'état du type d'événement de l'identificateur de l'événement et de la ressource.

L'enregistrement peut être effectué pour une seule ressource unique ou un groupe de ressources.

***Exemple* :**

Pour la métrique informationnelle Nombre d'interruptions du serveur X et en supposant que la source de données fournit une notification lorsqu'un serveur s'arrête ou démarre et que cette notification indique si le serveur est arrêté ou en marche à un moment donné, l'enregistrement se présente comme suit :

Type d'événement : Evénement de serveur en marche/arrêté

Ressource : Serveur X

Sur la base de l'enregistrement ci-dessus, le moteur filtre tous les événements comportant les définitions en marche/arrêté dans leur champ de type d'événement et dont le champ Ressource comporte Serveur X.

Une fois qu'un contrat est activé dans le système CA Business Service Insight, toutes les métriques enregistrent les événements pertinents nécessaires à leurs calculs. En fonction de ces demandes, le moteur de corrélation marque les événements pertinents à chaque logique applicative. Une fois que les calculs commencent, les événements pertinents sont envoyés à chaque métrique pour le calcul.

Ressources et gestion

Afin de permettre à l'enregistrement d'être dynamique, les ressources peuvent être allouées à titre individuel par leur nom/identifiant propre et unique ou par leurs relations avec les groupes logiques.

***Exemple* :**

Pour la métrique Nombre total d'interruptions sur les serveurs de centre de données**,** l'enregistrement est :

Type d'événement : Evénements en marche/arrêté

Ressource : Toutes les ressources étiquetées en tant que serveurs de centre de données. (Il s'agirait probablement d'un groupe de ressources.)

Présentation du cycle de vie des ressources

Une ressource est une entité physique ou logique qui peut modifier ses caractéristiques avec le temps. Elle peut être allouée à certains composants de service ou contractants, etc. à la fois, puis être réallouée à un moment futur. Chacune de ces modifications ou réallocations est capturée par CA Business Service Insight afin de pouvoir effectuer des calculs à tout moment, en fonction de la configuration des ressources et de l'affectation à ce moment précis.

Vous pouvez effectuer des modifications ou des réallocations d'une ressource à tout moment, mais cela nécessite la création d'une nouvelle version de cette ressource. Une date d'entrée en vigueur qui prendra effet lorsque des modifications se produiront doit être indiquée sur chaque nouvelle version. Les modifications seront alors reportées dans le futur, à moins que des modifications supplémentaires ne soient rencontrées dans une version ultérieure de cette même ressource. Toutes les modifications ne seront visibles et disponibles pour le moteur de calcul qu'une fois que cette nouvelle version a été activée et a pris effet. Ce processus est désigné sous le terme de "validation" de la ressource.

Dans CA Business Service Insight, il existe aussi une manière de gérer plusieurs allocations de ressources en une seule étape. Cette méthode consiste à utiliser un 'Ensemble de modifications'. Les ensembles de modifications permettent d'effectuer de grands volumes de modifications de ressources dans une transaction unique, de la même façon que fonctionne une base de données transactionnelle. Toutes les modifications peuvent être apportées à toutes les ressources qui sont allouées à un ensemble de modifications en effectuant l'opération sur l'ensemble de modifications dans son ensemble, puis en validant la modification en une seule étape.

Lorsqu'il s'agit des ressources et de leurs modifications, il est utile d'examiner les points suivants au niveau du moteur de calcul :

- Lors de l'activation (validation) de modifications sur une ressource spécifique ou sur un ensemble de ressources (ensemble de modifications), essayez d'envisager ce qui va être affecté dans le système. Etant donné que la modification du module de ressources est susceptible de déclencher un nouveau calcul, il est important d'optimiser la date d'activation de la ou des ressource(s) et le nombre de modifications effectuées en une seule opération.
- **Mise à jour en série** : Il est possible d'appliquer la même modification à de nombreuses ressources (mise à jour en série). Etant donné que la modification du module de ressources peut entraîner un nouveau calcul, il est important d'optimiser cette opération.

L'exemple précédent a résolu une ressource non pas directement, mais selon l'allocation logique à sa fonction ou sa localisation (dans ce cas, à sa fonction, un serveur de centre de données).

La demande d'enregistrement peut être très lourde si les événements sont demandés pour chaque serveur faisant partie du centre de données. Un des problèmes est le nombre de ressources auxquelles il est fait référence. L'autre est que l'infrastructure du centre de données est modifiée régulièrement. Ainsi, un serveur qui a fait partie du centre de données risque de ne plus être là ou un nouveau serveur peut être ajouté. Par conséquent, cette liste doit être dynamique.

D'après l'exemple précédent, il est clair que les ressources doivent être associées à un groupe logique afin d'être traitées via cette entité logique. En outre, le groupe logique risque de devoir être géré s'il change constamment.

L'allocation de ressources est la méthode CA Business Service Insight pour l'étiquetage des ressources. Vous pouvez allouer une ressource à un ou plusieurs groupes, types de ressource, contractants ou services. Les allocations de ressource sont gérées à l'aide du contrôle de version de CA Business Service Insight.

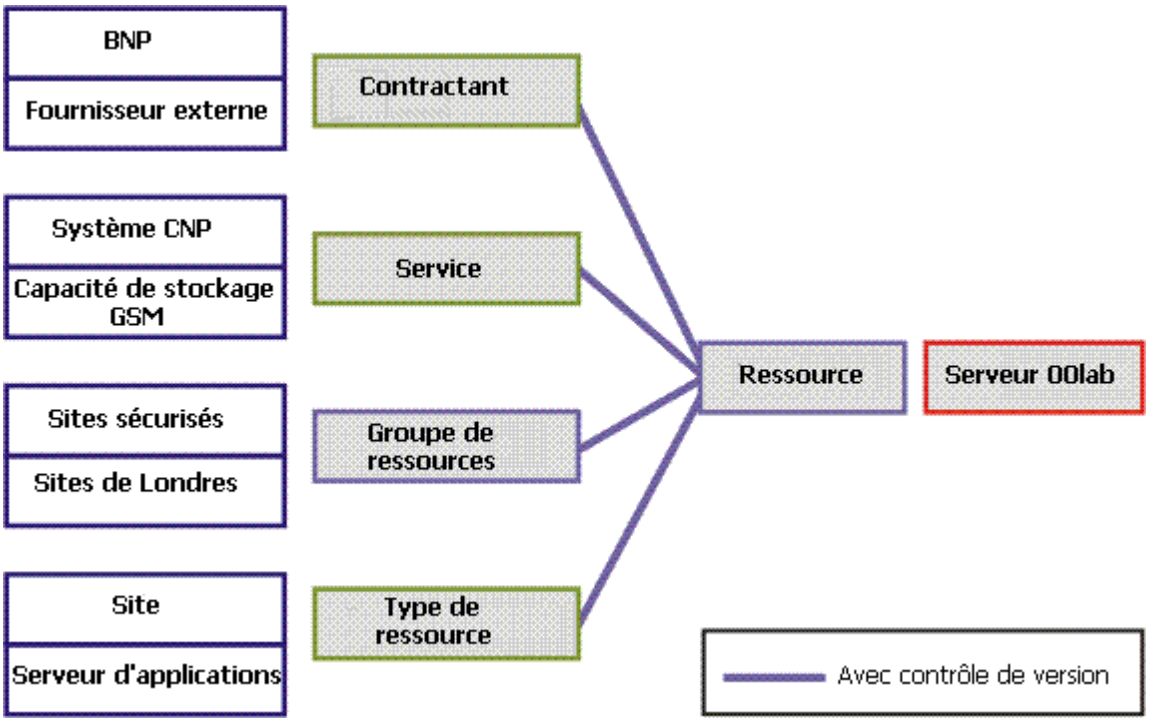
Les ressources disponibles pour l'inclusion dans les calculs sont déterminées par les ressources actuellement en vigueur dans le système (par rapport à l'intervalle de temps calculé à ce moment-là).

Maintenant, revenons à l'exemple précédent :

Nombre total d'interruptions sur les serveurs de centre de données

Vous pouvez représenter le centre de données dans le système comme un service auquel sont alors alloués tous les serveurs du centre de données. Il peut également être défini comme groupe de ressources intitulé "serveurs de centre de données". Il s'agit des deux méthodes que vous pouvez choisir pour l'allocation de ressources dans ce cas particulier, mais d'autres options sont disponibles.

Le diagramme suivant démontre à quelles entités une ressource peut être associée ainsi que ses utilisations logiques.



Un groupe de ressources peut refléter tous les aspects de la ressource requis pour les calculs, tels que la localisation ou la technologie qu'elle contient.

L'objectif principal de l'allocation de ressources à ces entités est d'assurer à la fois la correspondance avec les exigences de calcul et que le modèle reste aussi dynamique que possible.

Attributs personnalisés

Une autre caractéristique disponible pour une ressource est la capacité de fournir des attributs personnalisés. Ces attributs personnalisés peuvent être ajoutés à chaque type de ressource, et chaque ressource définie comme étant de ce type de ressource particulier, hérite de ces attributs personnalisés.

Reprenons l'exemple précédent, pour chacun des serveurs de centre de données, associez aussi leur adresse IP. Si chacun des serveurs de centre de données est défini avec le type de ressource Serveur de centre de données, veillez à ce qu'un attribut personnalisé appelé IP_Address soit ajouté au type de ressource de serveur de centre de données. Ainsi, vous pouvez alors associer chaque ressource (serveur) à son adresse IP via l'attribut personnalisé.

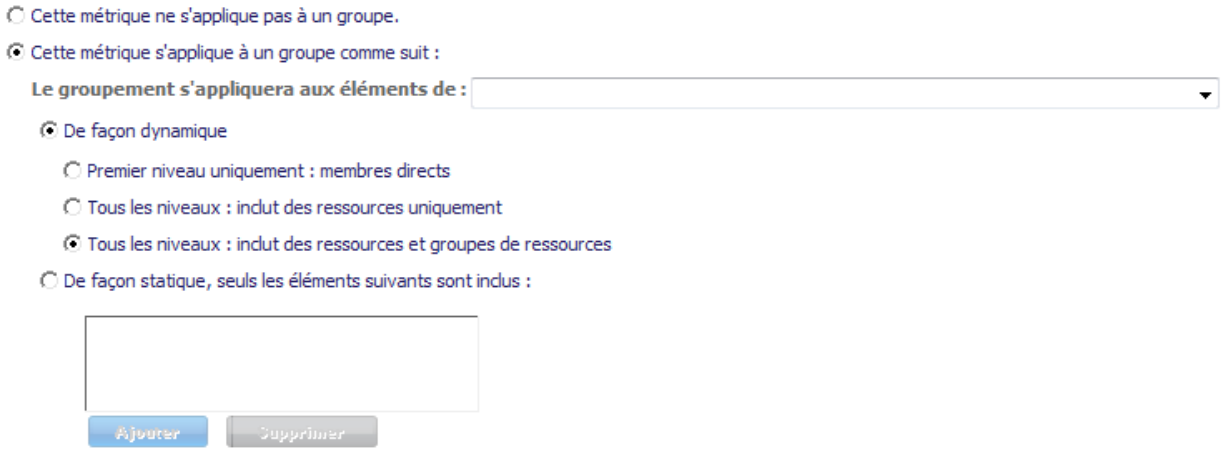
Remarque : Pour plus de détails et d'exemples d'étude de cas, voir la section Etude de cas - Utilisation des attributs personnalisés (page 304).

Qu'est-ce qu'une métrique groupée ?

Une métrique groupée est une métrique définie pour calculer les niveaux de service d'un groupe de ressources. Les calculs sont effectués pour chaque ressource individuellement, sans qu'il soit nécessaire de dupliquer la métrique à chaque fois avec l'enregistrement d'une ressource différente.

Le groupement ne peut s'effectuer qu'au niveau d'un groupe de ressources pour lesquelles il n'existe aucun niveau de service cible ou qui disposent du même niveau de service cible. Par exemple, la disponibilité de tous les serveurs d'applications doit être au moins égale à 99,9 % par serveur. Dans ce cas, une métrique est groupée au niveau d'un groupe de ressources qui contient tous les serveurs d'applications. Le moteur calcule un résultat de niveau de service distinct pour chaque serveur, la cible pour chaque serveur étant de 99,9 %.

Outre ce type de groupement, CA Business Service Insight prend en charge un type de groupement de "cumul" qui permet à une métrique unique de consigner des niveaux multiples de ressources (et de groupes). Le niveau de service est ainsi calculé à des niveaux multiples de la hiérarchie des ressources et une fonctionnalité d'exploration des niveaux supérieurs et inférieurs est applicable entre les entités liées à cette structure de ressources. A partir de l'onglet Groupement des métriques, vous pouvez activer cette option en sélectionnant les options pertinentes sur cette page.



Création du modèle de données du système

Dans le cadre du processus de modélisation des données, les composants obligatoires sont identifiés en fonction de la source de données et des besoins en calcul.

La liste suivante est une liste des composants qui devraient être identifiés dans le processus de modélisation des données ainsi que leurs définitions.

Nom d'événement
Nom d'événement tel qu'il apparaît dans CA Business Service Insight ; doit être aussi descriptif que possible.

Comportement de l'événement
Comportement d'un événement spécifié ; date à laquelle il a été reçu de la source de données, les conditions, etc.

Champ Horodatage
Champ de la source de données indiquant la date et l'heure d'un événement.

Champ Type d'événement
Champ de la source de données à convertir en type d'événement, et décrivant le type de rapport. Il est important que le nombre de types d'événement différents soit réduit au minimum dans la mesure du possible, car la définition du type d'événement s'effectue manuellement et idéalement cette opération ne devrait être effectuée qu'une seule fois.

Champs Données
Champs de la source de données à récupérer en tant que champs de données.

Champ Ressource
Champ de la source de données à convertir en ressource. Contient une entité à signaler avec un cycle de vie relativement fixe. La ressource est une entité avec un cycle de vie défini, où vous pouvez gérer des modifications dynamiquement au sein du système. Consulter un cycle de vie de ressource indique la fréquence à laquelle de nouvelles ressources sont ajoutées et modifiées en fonction de différents services ou d'une autre entité d'allocation, comme mentionné ci-dessus. Le champ à convertir en ressource dans CA Business Service Insight devrait comporter peu de changements d'allocation et des modifications limitées.

Et en dernier lieu, d'après toutes les définitions mentionnées ci-dessus :

Enregistrement par

Définit le critère d'enregistrement. Le critère d'enregistrement définit les critères Type d'événement et Ressource que la métrique enregistre. Une demande de ressources peut s'effectuer directement grâce à un enregistrement par Ressource, ou par entité d'allocation, comme Service, Contractant, Groupe de ressources, Type de ressource, et ainsi de suite. Cette définition est déterminée par les fonctionnalités d'enregistrement.

Une autre méthode consiste à utiliser l'enregistrement par métrique, où la métrique actuelle obtient des sorties (résultat de niveau de service) d'une autre métrique et les utilise comme entrée. Il est également possible d'utiliser le résultat de plusieurs métriques comme entrée.

Directives concernant la définition d'enregistrements

Utilisez les directives suivantes pour définir les enregistrements :

- Ne définissez jamais l'enregistrement par type d'événement uniquement. Même si la condition de calcul n'exige pas le filtrage des ressources, ajoutez au moins le filtrage des ressources sur le type de ressource.

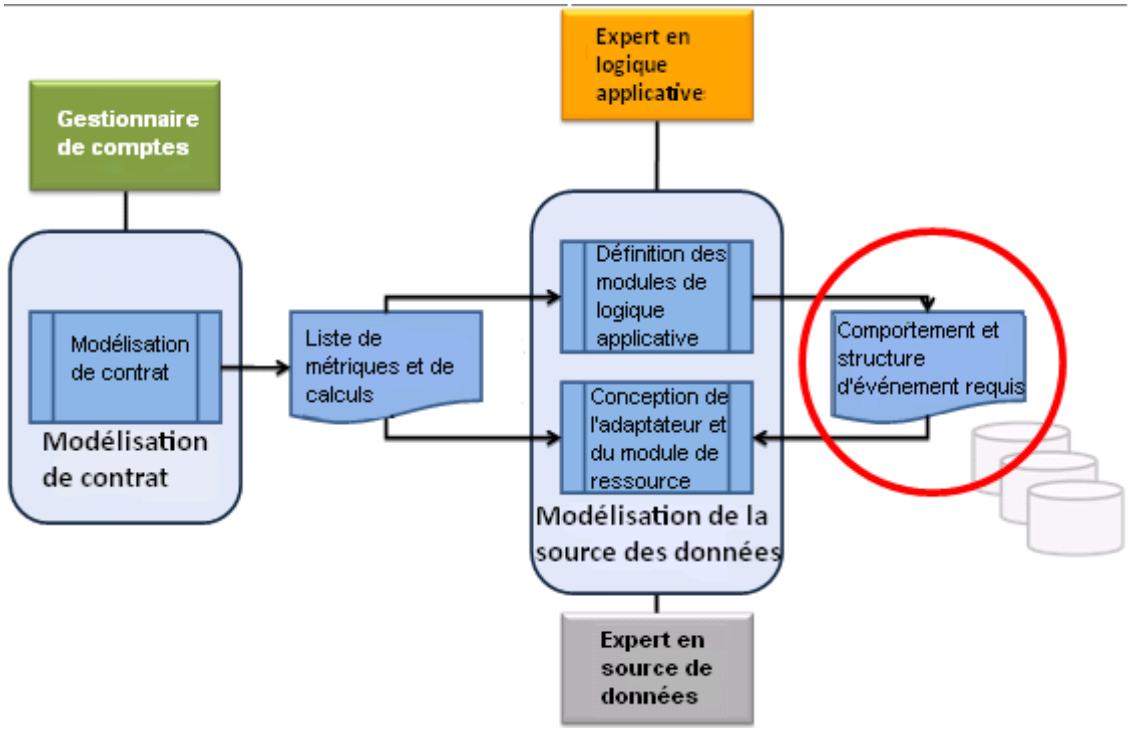
  Par exemple, dans le calcul des temps de réponse moyens globaux des serveurs d'application, l'événement Temps de réponse doit être signalé uniquement lorsqu'il est associé à l'un de ces serveurs d'application (c'est-à-dire dans le cas où le type de ressource de la ressource associée à l'événement est Serveur d'application). Dans un tel cas, le système peut éventuellement demander à d'autres types de ressource comme des sites ou des routeurs utilisant le même type d'événement d'envoyer des données et ce, afin de faire la distinction entre eux. Le type de ressource à signaler (Serveur d'application) devrait être ajouté à la commande d'enregistrement en tant que 3^{e} paramètre.

  Cela est dû au fait que lorsqu'une ressource est modifiée, le moteur indique que les métriques associées à cette ressource exigent un nouveau calcul. Si la métrique enregistre uniquement par type d'événement, le moteur affiche la métrique comme enregistrée à toutes les ressources et par conséquent indique qu'elle requiert un nouveau calcul lors de l'activation de chaque ressource. Pour éviter cela, vous devez utiliser le 3^{e} paramètre facultatif du type de ressource.

- La méthode la plus efficace pour l'enregistrement est l'utilisation de contractant et de service. En agençant les ressources de cette manière, il est possible d'exprimer la relation logique entre la couche de données et la couche d'activité professionnelle du système. L'enregistrement des ressources par le biais de ces entités permet de garder les formules intactes lorsqu'elles sont utilisées dans différents contrats ou pour différents services. Le contrat et le service de la métrique définissent le contractant et le service adaptés. Les formules de logique applicative définies dans ce type d'enregistrement sont facilement réutilisables parce qu'elles n'exigent aucune modification au cours de l'enregistrement.

Remarque : Les enregistrements peuvent également être traités dans l'onglet *Enregistrement* de chaque métrique. Cette interface fournit un assistant qui vous guide dans le processus pour la métrique sélectionnée.

Sorties de l'étape de modélisation de données (expert en sources de données et expert en logique applicative)



Afin de procéder à l'étape d'implémentation de la solution, l'expert en sources de données doit posséder les compétences suivantes :

- Une bonne compréhension des sources de données à manipuler, y compris :
  - Exemples de données historiques
  - Méthode de communication
  - Une compréhension des champs clé de chaque source de données à utiliser en tant que composants clés du modèle de ressource
- En effet, chaque source de données comporte une liste inhérente de types d'événement décrivant la structure et le comportement, y compris les définitions suivantes :
  - Nom d'événement
  - Comportement de l'événement
  - Champ Horodatage
  - Champ Type d'événement
  - Champs Données
  - Champ Ressource

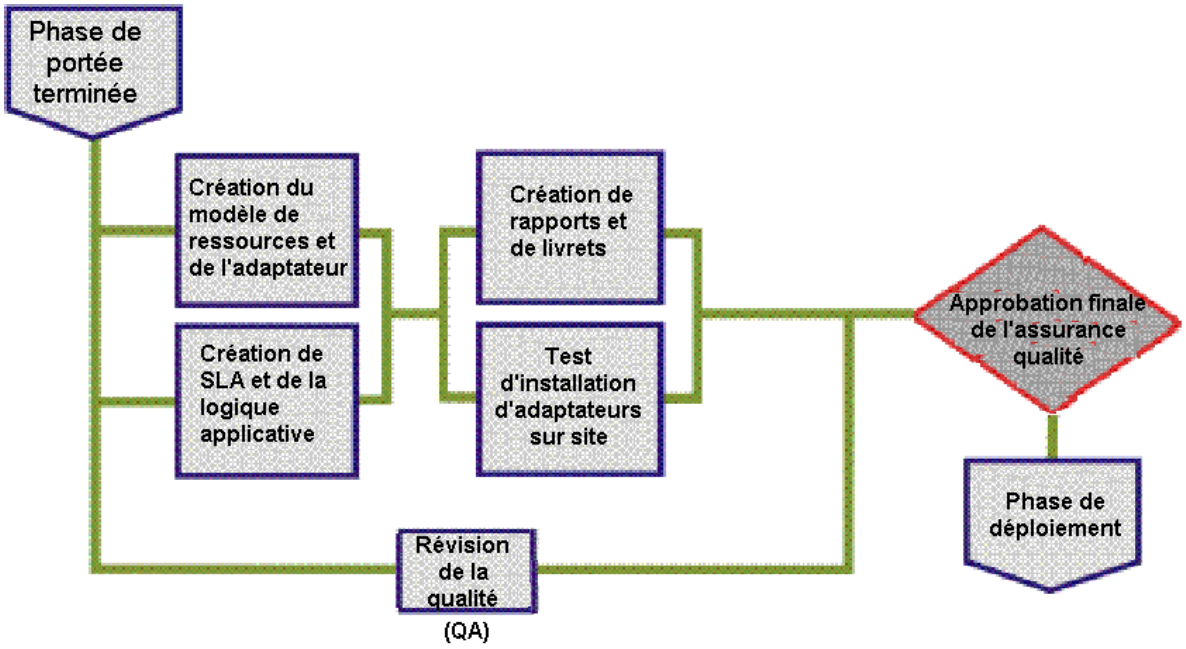
Remarque : Vous devriez créer un document récapitulatif des informations que vous acquérez.

Chapitre 3 : Implémentation

Ce chapitre traite des sujets suivants :

Implémentation : introduction (page 71)
Configuration de la structure (gestionnaire de contrats) (page 74)
Configuration de la bibliothèque de modèles (gestionnaire de contrats) (page 75)
Création de contrats (gestionnaire de contrats) (page 75)
Collecte de données (expert en sources de données) (page 82)
Ecriture de scripts de logique applicative (expert en logique applicative) (page 154)
Activation de contrats (gestionnaire de contrats) (page 200)
Création de livrables (gestionnaire de contrats) (page 204)

Implémentation : introduction



Ce chapitre explique le processus et le raisonnement sous-tendant la phase d'implémentation du projet. Comme présenté dans l'illustration précédente, la phase d'implémentation suit la phase de conception, qui est elle-même suivie par la phase d'installation et de déploiement.

L'objectif de la phase d'implémentation est que le gestionnaire de contrats puisse terminer la création de tous les éléments et les objets du système CA Business Service Insight, qui ont été définis pendant l'étape de conception. Pendant la phase d'implémentation, l'équipe prépare le système en vue de son déploiement et de son installation.

Cette phase ne doit pas avoir lieu avant que la phase de conception complète ne soit terminée et que tous les objectifs nécessaires aient été pris en compte et correctement définis. Si la phase de conception n'a pas été terminée correctement ou si tous les contrats, métriques, adaptateurs, etc. n'ont pas été clairement définis, le fonctionnement du système sera problématique ou ce dernier ne disposera pas de tous les éléments qui aurait dû être implémentés pendant la phase d'implémentation. La phase d'implémentation ne doit pas commencer avant la validation de la conception.

Il est également important que la phase d'implémentation soit terminée correctement avant de procéder à l'installation et au déploiement du système. Cette opération ne doit être effectuée qu'après un contrôle qualité.

Le processus de configuration implique que le gestionnaire de contrats se charge des étapes suivantes :

- Configuration des catalogues de services
- Création de contrats
- Activation des contrats
- Configuration des paramètres de sécurité
- Création de rapports, d'alertes et de tableaux de bord

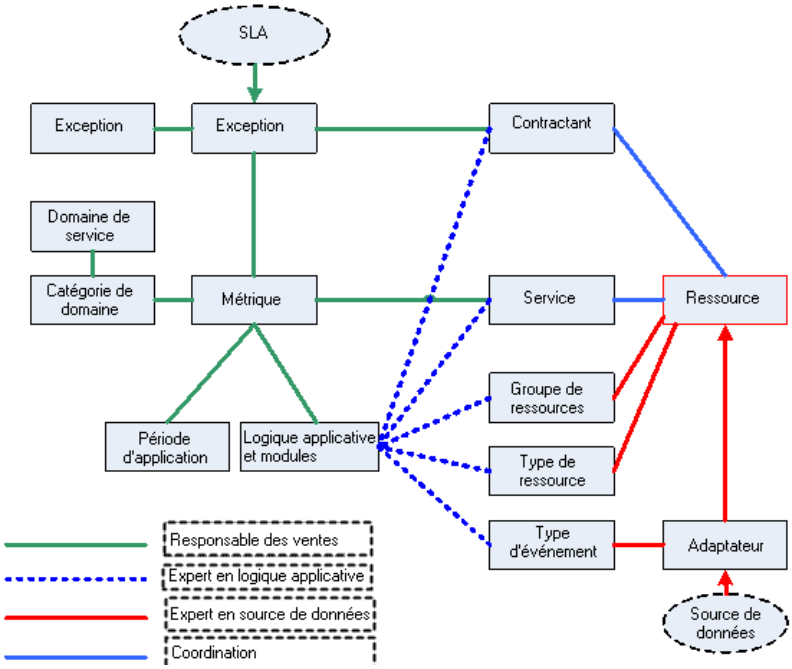
L'expert en sources de données doit procéder à la configuration de l'adaptateur et à son intégration aux sources de données. L'expert en sources de données doit également accomplir l'étape de conversion des ressources afin de lier les structures de sources de données aux entités définies dans le système CA. Cette étape est cruciale au niveau du processus global et peut également exiger les conseils du gestionnaire de contrats.

En outre, l'expert en logiques applicatives doit rédiger la logique applicative de toutes les métriques, en fonction des plans de la phase de conception. Celle-ci peut inclure la création de tous les modules obligatoires et la configuration des paramètres associés afin de fournir les fonctionnalités de calcul attendues.

Tous les points mentionnés ci-dessus sont expliqués en détail dans les sections de ce chapitre.

Remarque : Il est important que le gestionnaire de contrats soit conscient que tout mauvais choix fait à cette étape peut avoir des répercussions défavorables sur le fonctionnement de CA Business Service Insight et peut être difficile voire impossible à rectifier ultérieurement.

L'illustration suivante présente le flux de travaux logique global.



Configuration de la structure (gestionnaire de contrats)

La structure permet de :

- Définir des services, des groupes de service, des domaines de service et des unités
- Créer et conserver des modèles, y compris des modèles de logique applicative et de période d'application et des modules de logique applicative
- Gérer des attributs personnalisés pour les types de ressource

A cette étape, toutes les entités du système identifiées pendant la phase de conception sont créées dans la section relative à la structure de l'application. La création de contrats et de leurs métriques connexes n'est possible que lorsque le système contient ces entités structurelles.

La création de la structure implique l'ajout des nouveaux éléments suivants :

- Services
- Groupes de services
- Domaines de service et catégories de domaine
- Unités de mesure
- Modèles de période d'application
- Contractants
- Attributs personnalisés

Pour plus d'informations sur chacun des éléments mentionnés ci-dessus, consultez l'aide en ligne.

Configuration de la bibliothèque de modèles (gestionnaire de contrats)

La bibliothèque de modèles permet de définir et de gérer les éléments suivants :

- Bibliothèques de modèles
- Dossiers Modèle
- Modèles de niveau de service
- Modèles de contrats
- Paramètres de sécurité relatifs aux droits d'accès utilisateur

Pour plus d'informations sur chacun des éléments mentionnés ci-dessus, consultez l'aide en ligne.

Création de contrats (gestionnaire de contrats)

A cette étape, les contrats et leurs entités connexes définis durant la phase de conception sont créés dans le système.

Procédez comme suit :

1. Ajoutez un nouveau contrat et appliquez les détails généraux du contrat.
2. Pour chaque contrat, définissez ses métriques et appliquez leurs informations générales.

Seules les informations générales du contenu d'un contrat sont appliquées à cette étape ; la logique applicative et le groupement des métriques du contrat sont exclus.

La description des étapes qui suit met l'accent sur certains points importants devant être considérés à cette étape. Ces étapes sont décrites en détail dans l'aide en ligne.

Etape 1 : Ajoutez un nouveau contrat et appliquez les détails généraux du contrat.

La définition du contrat doit inclure :

- La définition du nom du contrat.
- La sélection du ou des contractants connexes.
- L'association des services connexes.

- La définition des dates d'entrée en vigueur pour le Contrat. Les dates d'entrée en vigueur des contrats correspondent à la plage de dates pour laquelle le moteur de corrélation calcule le niveau de service de ce contrat ; les résultats de rapport sont disponibles uniquement pour ces dates. Lors de la définition des dates, il est important de tenir compte des conditions relatives à la disponibilité des rapports connexes du contrat et aux données brutes disponibles.
- Définition du fuseau horaire et de la devise du contrat. Cette définition a pour but la génération de rapports et elle permet de générer des rapports sur ce contrat en fonction du fuseau horaire pertinent. La définition de la devise permet au moteur de génération de rapports de déterminer dans quelle devise les pénalités exprimées par les formules de pénalité doivent être affichés.

Etape 2 : Ajoutez les détails généraux des métriques.

Une fois que le contrat est en place, il est temps de créer ses métriques. Au cours de la définition des métriques, les étapes suivantes doivent être exécutées :

- Définition du nom de la métrique.
- Application des détails de la métrique (Service, Domaine, Unité, Période d'application, Fuseau horaire, etc.).
- Définition des seuils du tableau de bord (consultez Configuration des Pages du tableau de bord (page 218)).
- Association des métriques connexes (le cas échéant) et leur lien de relation.
- Définition de la granularité du calcul de la métrique par le moteur.
- Définition de l'énoncé d'objectif et des paramètres.

A cette étape, la définition de la métrique ne comporte pas encore les éléments suivants :

- Formule/module de logique applicative et enregistrement
- Définition du groupement

Ces éléments sont inclus dans les métriques du contrat uniquement après que l'infrastructure de système a été créée et que les formules de logique applicative ont été développées et testées.

Remarque : Une alternative à l'approche mentionnée ci-dessus consiste à développer d'abord les modèles de niveau de service dans le système, plutôt qu'à définir les contrats tout de suite. Cela vous permet de créer le modèle que vous pouvez alors utiliser pour créer d'autres contrats. Dans certains cas, vous pouvez importer les modèles de niveau de service préexistants dans le système à partir d'une autre instance de CA. Vous pouvez prendre ainsi une longueur d'avance par rapport à la création d'un contrat à partir de zéro. Pour plus d'informations sur la création de vos propres modèles de niveau de service, consultez à la section Créer des modèles de niveau de service (page 79).

Dans certains cas, il est recommandé de créer un exemple de contrat initialement, afin de vérifier que tout fonctionne correctement et comme prévu. Il est alors possible de créer le ou les modèles de niveau de ce contrat et de les stocker dans le catalogue des services. Vous constituez ainsi un point de départ solide pour tous les contrats créés dans le système.

Créer des contrats à partir d'un service

La création de contrats dans le système à l'aide d'un catalogue des services, avec ou sans modèle (selon plusieurs modèles de niveau de service), est un moyen beaucoup plus efficace ; le contrat peut être réutilisé plus facilement et il s'applique à un grand nombre de contrats divers. En général, les modèles de niveau de service contiennent un ensemble de métriques prédéfini applicable à certains composants de service. Si nécessaire, vous pouvez également avoir plus d'un service (et les métriques associées) dans un niveau de service unique. Généralement, le contenu d'une définition de service est défini en fonction de son utilisation, et il peut varier selon les exigences.

Par exemple, considérons un service d'hébergement d'application proposé par une organisation. L'organisation peut offrir à ses clients trois packages de services différents comme Bronze, Argent (Silver) et Or (Gold), selon ce qui est inclus dans le package. Un bon exemple de l'utilisation de modèles de niveau de service consisterait éventuellement à en créer un pour chacun des packages.

Une fois déterminées, vous pouvez utiliser ces définitions pour créer un nouveau contrat de client très efficacement. Par exemple, le client ABC décide de prendre un package d'hébergement d'application Or (Gold). Vous pouvez créer ce nouveau contrat dans le système en utilisant directement la définition de service comme suit :

Dans la page Contrats, cliquez sur Ajouter, Utilisation du catalogue des services, puis sélectionnez D'après le modèle ou Sans modèle. Suivez alors l'assistant de création de contrats pour finaliser la création du contrat. Si vous choisissez D'après le modèle, vous devez alors spécifier les paramètres de modèle.

L'assistant de création de contrats affiche une liste des modèles de niveau de service et vous pouvez spécifier ceux à inclure dans le contrat. Notez qu'il est possible de sélectionner des métriques spécifiques parmi plusieurs modèles de niveau de service, ou tout simplement de sélectionner la définition entière. Dans cet exemple, toutes les métriques du package d'hébergement Or (Gold). Veuillez noter qu'en sélectionnant le niveau le plus élevé, tous les noeuds enfants sont automatiquement sélectionnés. Notez également qu'il est possible d'affecter les métriques à un service différent si nécessaire. Toutefois, elles sont, par défaut, assignées aux mêmes composants de service que ceux de la définition.

Une fois que toutes les métriques requises sont sélectionnées, cliquez sur le bouton Suivant pour transférer ces métriques vers le nouveau modèle de contrat et entrer un nom de contrat et des détails généraux. Cliquez sur Enregistrer et continuer pour créer le contrat.

Une fois que cela est terminé, vous disposez des options suivantes :

- Poursuivez les étapes de l'assistant de création de contrats, définissez des paramètres et exécutez l'assistant de modification de métriques pour ouvrir l'interface et personnaliser les métriques des contrats. L'assistant vous permet de contrôler et de modifier les métriques en modifiant les champs disponibles, comme les paramètres de métrique dans l'énoncé d'objectif, le nom de la métrique, la période d'application et la description. Une fois que chaque métrique a été passée en revue, l'assistant vous renvoie à la page Métriques de contrat, où vous pouvez fermer et enregistrer le nouveau contrat.
- Ouvrez la page Contrat pour afficher ou modifier le contrat.

Créer des modèles de niveau de service

La création d'un modèle de niveau de service est une opération assez simple. A partir d'un contrat existant (sur la page Détails du contrat), sélectionnez toutes les métriques que vous souhaitez inclure et cliquez sur Enregistrer comme modèle de niveau de service.

La fenêtre suivante vous invite à nommer le modèle de niveau de service. Vous pouvez alors enregistrer le modèle de niveau de service. Une fois enregistrées, toutes les périodes d'application associées aux métriques sélectionnées sont incluses dans l'onglet Période d'application. A partir de là, il peut être nécessaire de personnaliser davantage le modèle de niveau de service afin de la rendre complètement flexible, notamment grâce à l'ajout de paramètres aux métriques et à la présentation des paramètres de chaque métrique via l'énoncé d'objectif. Cela peut aussi impliquer la création de paramètres au niveau du modèle de niveau de service (similaires aux paramètres de contrat), qui peuvent alors être référencés par toutes les métriques ou quelques-unes seulement. Une fois terminé, le modèle de niveau de service est prêt à être déployé sur d'autres contrats.

Cycle de vie du contrat et méthodes de contrôle de version

Une fois qu'un contrat a été configuré complètement, il doit être validé. Cette action permet au moteur de calcul de commencer à calculer les niveaux de service de toutes les métriques du contrat. En validant le contrat, son statut passe de Préliminaire (modifiable) à En vigueur (non modifiable). Toute autre modification requise sur ce contrat exige la création d'une nouvelle version. Si les mêmes dates d'entrée en vigueur sont choisies pour la nouvelle version, celle-ci écrase complètement la version précédente une fois que les modifications sont effectuées et que la nouvelle version est validée. Le moteur commence alors à recalculer les métriques qui sont différentes de celles présentes dans la version précédente. Des versions en vigueur peuvent également partiellement se chevaucher, par exemple lorsque les cibles de certaines métriques sont modifiées dans le contrat au milieu de la période en vigueur. Dans ce cas, l'ancienne version est utilisée jusqu'à ce que la date d'entrée en vigueur de la deuxième version prenne effet. La deuxième version prend alors le statut En vigueur lors des calculs.

Le tableau suivant présente les modifications effectuées par l'utilisateur par rapport à la portée de l'impact du nouveau calcul et au délai. Une modification peut affecter soit une métrique dans une version de contrat précise ou des métriques qui s'appliquent à plusieurs contrats et des versions de contrat.

Changement	Portée de l'impact	Délai d'impact
Modifications apportées aux métriques		
Modification des détails de métrique : formule de logique applicative	Affecte toutes les métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Modification des détails de métrique : valeur cible	Affecte toutes les métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Modification des détails de métrique : période de suivi	Affecte toutes les métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Modification des détails de métrique : paramètres de métriques	Affecte toutes les métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Modification des détails de métrique : fuseau horaire	Affecte toutes les métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Modification des détails de métrique : groupement	Affecte toutes les métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat

Modification des détails de métrique : dates d'entrée en vigueur	Affecte toutes les métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Modification des détails de métrique : périodes d'application	Affecte toutes les métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Modification des paramètres de contrat	Affecte toutes les Métriques dans la version spécifique de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Modification des modules SLALOM	Affecte toutes les métriques liées au module slalom modifié, dans tous les contrats et versions de contrat	Depuis la date d'entrée en vigueur de la version du contrat
Opérations générales		
Modification du modèle de ressource de la métrique/de l'ensemble de modifications (CA Business Service Insight 4.0)	Affecte toutes les métriques qui enregistrent la ressource, dans l'ensemble des contrats et versions de contrat	Depuis l'état le plus proche précédant la date à laquelle une modification de ressource est survenue.
Obtention d'événements retardés pour la métrique Evénements dotés d'un horodatage passé (données brutes ou événements réutilisables)	Affecte toutes les métriques qui utilisent des ressources, dans l'ensemble des contrats et versions de contrat	Depuis l'état le plus proche précédant la date à laquelle une modification d'événement est survenue.
Données de correction relatives à l'ajout d'une métrique	Affecte toutes les métriques qui utilisent des ressources, dans l'ensemble des contrats et versions de contrat	Depuis l'état le plus proche précédant la date à laquelle une modification d'événement est survenue.
Modifier, mettre à jour ou supprimer une période d'exception de métrique (activation et désactivation)	Selon les paramètres d'exception et l'implémentation spécifique, affecte une métrique spécifique dans une version de contrat précise ou peut affecter des métriques qui s'appliquent à plusieurs contrats et versions de contrat.	Aussi proche que possible de la période d'exception

Mise à jour de l'attribut personnalisé	Affecte toutes les métriques qui enregistrent la ressource, dans l'ensemble des contrats et versions de contrat	Depuis l'état le plus proche précédant la date à laquelle une modification de ressource est survenue.

Enfin, voici quelques points clés à retenir à propos des versions de contrat :

- Si la nouvelle version a la même date d'entrée en vigueur, seules les métriques qui ont été changées sont recalculées, et elles sont recalculées depuis le début de la version.
- Si la nouvelle version a des dates différentes, toutes les métriques de la nouvelle version sont calculées depuis le début de cette version et toutes les métriques de la version précédente sont recalculées à partir d'un certain point dans cette version jusqu'à ce que la nouvelle version devienne valide. Le nombre de calculs dépend de la configuration des états.
- Il est recommandé de créer des contrats en vigueur pendant 1 an et de les renouveler lorsqu'ils arrivent à expiration, afin d'éviter des périodes de calcul excédant un an.
- Les versions de contrat qui ne sont pas en vigueur (la date actuelle est postérieure aux dates d'entrée en vigueur du contrat) sont calculées, et elles sont donc considérées comme des métriques actives dans le système, puisqu'elles servent au calcul des données de niveau de service qui leur sont associées en vue de la génération de rapports.

Les valeurs associées à des variables globales au niveau des métriques ne sont pas utilisées par plusieurs versions de contrat (c'est-à-dire, la routine OnLoad de la logique applicative est invoquée au début de chaque version de contrat).

Remarque : Pour obtenir un certain nombre de scénarios structurés et d'études de cas, consultez la section Etudes de cas de modélisation de contrats (page 271).

Collecte de données (expert en sources de données)

Au cours de la phase de collecte de données du processus d'implémentation, vous utilisez des adaptateurs. Les discussions suivantes expliquent ce processus.

Fonctionnalité d'adaptateur

Les adaptateurs sont des modules chargés de rassembler les données des sources de données et de les transférer au système de CA Business Service Insight. Les adaptateurs filtrent des données venant de la source de données et les manipulent de sorte qu'elles ne contiennent que les informations requises pour les calculs de niveau de service dans la structure correcte, une fois qu'elles arrivent dans le système.

La plate-forme d'adaptateur permet de :

- Recevoir des données en ligne ou hors ligne quelle que soit la fréquence requise
- Recevoir des données à divers niveaux : brutes, calculées ou agrégées
- Recevoir des données provenant d'une vaste gamme de types d'outils

Dans l'ensemble, tous les adaptateurs sont dotés de deux composants :

- **Composant d'adaptateur générique** :

  Il existe deux types de composant d'adaptateur générique : un composant d'adaptateur de fichier ASCII et un composant d'adaptateur de type SQL ODBC. Ces composants permettent de se connecter à une source de données et d'en faire l'analyse en tant que fichier ASCII ou d'exécuter une requête SQL sur cette dernière.

- **Fichier de configuration de l'adaptateur** :

  Tous les adaptateurs requièrent un fichier de configuration pour savoir où et comment procéder à la connexion, quelles données récupérer et comment transformer et convertir ces données en transactions et événements CA génériques. CA Business Service Insight dote tous les types d'adaptateurs génériques d'un fichier modèle de configuration XML par défaut qu'il est possible d'ajuster en fonction des détails spécifiques au client concernant la source de données à laquelle il doit se connecter. Le fichier de configuration XML définit quels champs doivent être récupérés, comment ils doivent être identifiés, comment ils doivent être convertis dans la base de données normalisée du système, etc.

  Remarque : Un Assistant de configuration d'adaptateur est intégré à l'interface utilisateur, permettant ainsi une personnalisation de base de ce modèle XML en ligne. L'objectif est le même que si vous créez le fichier de configuration XML pour l'adaptateur. Vous pouvez trouver davantage de détails sur cette fonctionnalité plus loin dans ce chapitre.

La plate-forme de l'adaptateur comprend un mécanisme de redémarrage/récupération et peut prendre en charge les problèmes concernant les données transmises par des outils tiers (pannes informatiques, problèmes réseau, données manquantes, duplication des données, données erronées, écarts dans les données, validation des données, etc.). Tous les adaptateurs prévoient l'intégrité des données et le suivi et la journalisation complets de tous les messages d'adaptateur ; ils sont abordés de façon détaillée plus loin dans ce manuel.

Les adaptateurs de CA Business Service Insight peuvent fonctionner en tant que service ou en tant qu'application (visible ou non visible). La technologie d'adaptateur de CA Business Service Insight prend en charge des mécanismes de sécurité avancée tels que le chiffrement, le protocole de transfert et les processus d'authentification.

Il est important de noter à cette étape que l'assistant de configuration d'adaptateur est un mécanisme automatisant les processus et les tâches suivants. Si certains éléments mentionnés ne sont pas forcément toujours visibles lorsque vous utilisez l'assistant, ils sont toutefois présents « dans les coulisses » de l'interface de l'assistant.

Environnement d'adaptateur

Les entités suivantes se rapportent à l'adaptateur et ses paramètres de configuration et d'exécution.

Source de données **:**

Source de données à laquelle l'adaptateur se connecte pour récupérer des données dans leur format d'origine.

Fichiers de travail **:**

Fichiers de sortie produits par l'adaptateur et écrits dans les processus (pour plus d'informations, consultez la section Fichiers de travail (page 89)).

Ecouteur d'adaptateur de CA Business Service Insight **:**

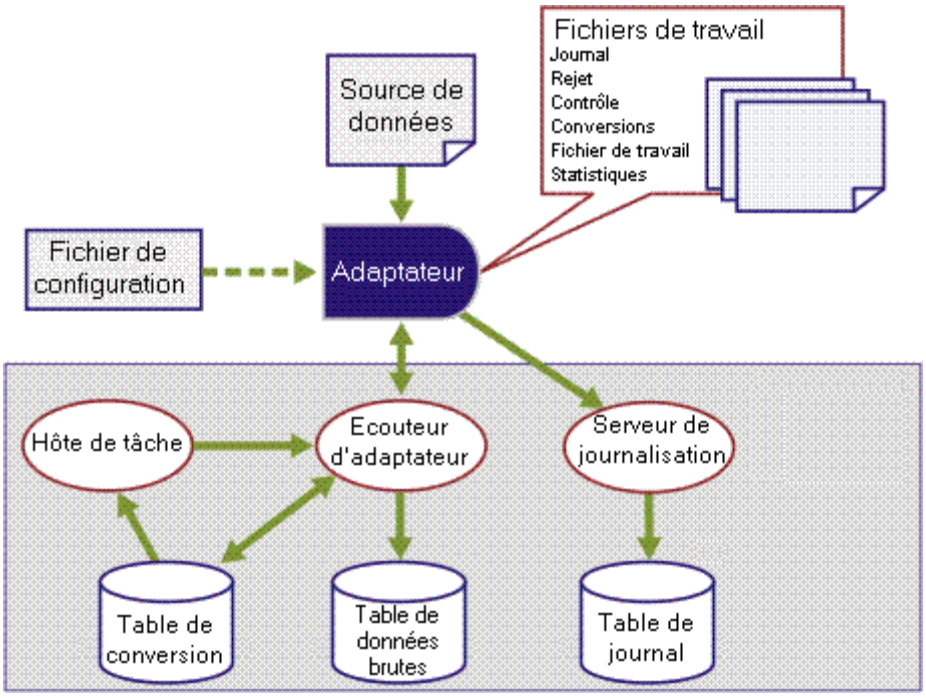
Trois types de message sont transférés entre l'adaptateur et l'écouteur d'adaptateur :

- **Contrôle** : messages de démarrage, d'arrêt et de mise en pause envoyés par l'écouteur à l'adaptateur et renvoyés lorsque l'adaptateur change de statut.
- **Conversions** : l'adaptateur envoie des demandes relatives au contenu de la table de conversion ainsi que des demandes relatives à des valeurs de conversion spécifiques. L'écouteur renvoie les tables et la valeur convertie. L'écouteur d'adaptateur reçoit une notification de l'hôte de tâche selon laquelle une entrée de conversion a été convertie. Il envoie alors le message à l'adaptateur.
- **Données brutes** : événements de données brutes unifiées et envoyées par l'adaptateur. Ces événements sont envoyés par paquets et incluent des messages d'acquittement.

Serveur de journalisation de CA Business Service Insight

Vous pouvez configurer l'adaptateur de sorte qu'il envoie des messages de journal au journal du système et qu'il les consigne dans un fichier local. Si le port et l'adresse IP du serveur de journalisation sont spécifiés et définis dans les paramètres de registre de l'adaptateur, alors l'adaptateur envoie aussi des messages au serveur de journalisation.

Le diagramme suivant décrit le processus de l'adaptateur dans la relation avec chaque entité avec laquelle il interagit.



Vous trouverez ci-dessous une description de l'interaction du processus de d'adaptateur avec ces entités :

Fichier de configuration **:**

Contient des paramètres pour tous les paramètres de configuration de l'adaptateur ou certains seulement. L'adaptateur utilise le fichier de configuration pour déterminer la méthode de connexion utilisée par l'adaptateur et les métriques nécessaires à l'analyse afin de créer la sortie d'événement. Il s'agit d'un fichier XML, et le format contient six éléments de base :

Général :

Attributs d'adaptateur variés (répertoire actif, fichiers de sortie et indicateur de débogage).

OblicoreInterface :

Attributs pour la connexion avec le serveur CA Business Service Insight.

DataSourceInterface :

Attributs utilisés pour la connexion avec la source de données (chemin d'accès de fichiers et modèle, chaînes de connexion, requêtes SQL, etc.)

InputFormatCollection :

Métrique d'analyse permettant d'analyser et de manipuler des sources de données.

TranslatorCollection :

Métrique permettant de créer l'événement unifié composé des champs de données analysés et manipulés.

TranslationTableCollection :

Métrique permettant de mapper des données entre des données d'origine et des entités de CA Business Service Insight.

Chacune de ces six sections contient toutes les informations pertinentes qui permettent à l'adaptateur de se connecter à la source de données, de récupérer les informations requises, d'en faire leur analyse dans les structures d'événements unifiés de CA Business Service Insight et de les stocker dans la table de données brutes de CA Business Service Insight.

Fichiers principaux

L'adaptateur compte deux fichiers principaux : le fichier exécutable et les fichiers de configuration. Le fichier exécutable est un fichier générique. Il existe en fait deux fichiers exécutables : l'adaptateur SQL et l'adaptateur de fichier.

Un fichier de configuration XML est adapté à chaque adaptateur afin de stocker les sources de données requises. Le fichier de configuration spécifie les informations relatives à la source de données (nom, emplacement, méthode de connexion et structure) et la structure des événements de sortie qui doivent être générés par l'adaptateur.

Le fichier de configuration inclut ces paramètres et ces valeurs définis en tant qu'attributs dans un fichier XML structuré prédéfini.

Lors de la création d'un nouvel adaptateur, il est nécessaire d'utiliser l'exécutable pertinent existant (d'après le type de source de données cible, fichier pour des sources de données de fichier plat, SQL pour des sources de données de base de données), puis de modifier le fichier de configuration selon vos besoins. Les deux structures contiennent des éléments de configuration légèrement différents selon qu'il s'agit d'un adaptateur SQL ou de texte. En règle générale, la configuration s'effectue automatiquement lors de la création de l'adaptateur à l'aide de l'utilitaire Gestionnaire d'adaptateurs.

Les autres fichiers liés aux adaptateurs sont des fichiers de travail créés par l'adaptateur lors de la lecture de la source de données et de l'écriture d'événements dans le système de CA Business Service Insight.

Remarque : Pour plus d'informations sur la modification du fichier de configuration, consultez Spécifications de configuration de l'adaptateur (page 377).

Fichiers de travail

Les fichiers de travail de l'adaptateur sont créés lorsque ce dernier est exécuté pour la première fois et ils sont mis à jour constamment à chaque exécution de l'adaptateur.

Chaque adaptateur dispose de ses propres fichiers de travail. Vous pouvez définir les noms des fichiers de travail dans le fichier de configuration de l'adaptateur (facultatif) ou conserver leurs noms par défaut. L'emplacement du fichier de travail est défini par le dossier de travail et peut également être défini dans le fichier de configuration. Notez que le chemin d'accès spécifié peut être relatif au répertoire actuel dans lequel l'adaptateur se trouve. Le chemin d'accès spécifié doit déjà exister (ou vous devez le créer) pour que l'adaptateur s'exécute correctement.

Remarque : Le dossier n'est pas créé automatiquement s'il n'existe pas.

Tous les paramètres pertinents dans le fichier de configuration sont contenus dans la section Général. Seul l'emplacement du fichier journal est placé dans le registre ou transféré via la ligne de commande.

AdapterStatistics.txt

L'adaptateur consigne des informations statistiques dans ce fichier à une minute d'intervalle. La dernière ligne est écrite lorsque l'adaptateur s'arrête. Chaque ligne dans le fichier contient le nombre d'éléments suivants :

- Evénements reçus
- Evénements ignorés
- Evénements avec des erreurs
- Messages envoyés
- Packages envoyés

Chaque fois que l'adaptateur est exécuté, il initie les statistiques.

RejectedEvents.txt

Ce fichier contient tous les événements que l'adaptateur n'a pas réussi à envoyer à CA Business Service Insight, car aucun ID dans la table de conversion ne correspond à la valeur d'événement définie comme nécessitant une conversion. En d'autres termes, la conversion appropriée n'a pas été effectuée. Chaque événement doté d'au moins une valeur en attente de conversion est écrit dans le fichier rejectedEvents.

Au début de chaque exécution, l'adaptateur tente d'abord d'envoyer à CA Business Service Insight les événements du fichier rejectedEvents tout en recherchant l'ID correspondant à la valeur pertinente dans la table de conversion. S'il trouve la valeur, l'adaptateur envoie l'événement et le supprime du fichier. Si une valeur correspondante est introuvable, l'événement reste dans le fichier rejectedEvents.

Il est possible de configurer le nombre maximum d'événements rejetés que vous pouvez atteindre en définissant le paramètre RejectedEventsUpperLimit dans le fichier de configuration de l'adaptateur. Lorsque la limite est atteinte, l'adaptateur arrête de lire de nouveaux enregistrements et passe au statut Bloqué. Vous pouvez vous en rendre compte lorsque la sortie de débogage s'affiche à l'écran pendant l'exécution de l'adaptateur. Si une chaîne continue de B majuscules s'affiche, l'adaptateur est alors bloqué ; certaines entrées en attente doivent être converties avant qu'il ne charge davantage de données.

Les événements en attente sont écrits dans le fichier au format XML. Exemple d'un événement du fichier :

```
<rejectedEvent
        createDate="1062330841"
        translator="Translator">
        <event
        inputFormat="InputFormat">
                <field name="resource" type="3" value="Server333p"/>
                <field name="timestamp" type="4" value="1036108800"/>
                <field name="memory_utilization" type="2" value="26.71"/>
                <field name="cpu_utilization" type="2" value="78.85"/>
        </event>
</rejectedEvent>
```

Journal d'adaptateur

Le journal d'adaptateur est le fichier dans lequel l'adaptateur écrit des messages de journal.

Il est recommandé de parcourir le fichier journal de l'adaptateur à l'aide de l'utilitaire de navigation de journal.

Il est possible de définir le niveau de données du rapport dans ce fichier journal en modifiant un paramètre du fichier de configuration de l'adaptateur : LogDebugMode. Lorsqu'il est défini sur Oui, l'adaptateur consigne des messages d'indication normaux dans le journal, ainsi que l'enregistrement d'origine, les résultats de l'analyse et l'événement choisi.

Ce paramètre est habituellement défini sur Oui pendant les opérations de test et de surveillance de l'adaptateur.

Par défaut, la taille du fichier n'excède pas 1 Mo. Lorsque cette limite est atteinte, l'adaptateur change de nom en y ajoutant « _old » et crée un nouveau fichier journal. L'adaptateur peut potentiellement stocker jusqu'à 2 Mo de messages de journal ; 1 Mo pour l'ancien fichier et 1 Mo pour le fichier actuel.

Vous pouvez configurer la taille limite du fichier journal en tant qu'entrée dans le registre pour chaque adaptateur avec un maximum de 10 Mo. L'entrée dans le registre est nommée LogFileMaxSize et elle est définie sous l'adaptateur spécifique, sa valeur étant un multiple de Ko.

DataSourceControl.xml

Le fichier DataSourceControl.xml est utilisé par l'adaptateur pour contrôler son accès à la source de données et s'assurer que quand il s'exécute, il continue toujours à partir du dernier point lu.

L'adaptateur de fichier conserve le nom du dernier fichier lu, la dernière ligne lue et la position qu'il a atteinte dans le fichier. Lorsque l'adaptateur est exécuté la fois suivante, il accède au fichier à l'emplacement voulu grâce aux informations trouvées dans le fichier DataSourceControl.xml. Grâce à ce mécanisme, l'adaptateur lit uniquement de nouveaux enregistrements à chaque exécution.

L'adaptateur ne fonctionne pas directement sur les fichiers sources ; il copie d'abord fichier actuel dans un fichier de travail. Les mêmes informations sont donc gardées pour le fichier de travail et pour le fichier source. Si le fichier source est modifié, seuls les enregistrements nouveaux sont copiés dans le fichier de travail.

Si l'adaptateur est configuré pour supprimer le fichier source une fois traité, en définissant le paramètre dans le fichier de configuration DeleteAfterProcessing sur Oui, il n'enregistre pas les informations dans le fichier source. Une fois fini, il lit tout nouveau fichier présent dans le dossier de travail qui correspond au modèle de fichier défini dans le fichier de configuration.

Il ne recherche de nouveaux enregistrements dans le dernier fichier que lorsque DeleteAfterProcessing est défini sur Non. S'il n'y en a aucun, il passe au fichier suivant dans un ordre lexicographique. Par conséquent, essayez de nommer vos fichiers de données sources selon un ordre croissant séquentiel pour vous assurer qu'ils sont lus selon la séquence correcte. Par exemple, ajoutez aux fichiers une valeur de date inverse (aaaammjj-hhmmss) à cet effet. Par exemple :

- DataSourceABC20070517-14:00:00.csv
- DataSourceABC20070517-15:30:00.csv
- DataSourceABC20070517-17:00:00.csv et ainsi de suite.

Ceci est un exemple de DataSourceControl.xml pour un adaptateur de fichier.

```
<AdapterControl Save="end" LastSaveTime="2005/05/20 13:06:39">
<Data><WorkData><LineNumber>0</LineNumber>
<FilePattern>c:\adapters\callsadapter\*adapterpca.csv</FilePattern>
[set the File Name variable]</FileName>
<BasicPosition>0</BasicPosition>
</WorkData>
<NonDeletedFiles>
```

```
<File NamePattern="c:\adapters\callsadapter\*adapterpca.csv">
[set the File Name variable]2005adapterpca.csv</FileName>
<LastLine>25/04/2005,5925,NN4B,12,12,0,10,0,11</LastLine>
<LastPosition>15427</LastPosition></File>
</NonDeletedFiles>
</Data>
</AdapterControl>
```

Un adaptateur SQL conserve la dernière valeur des champs clés de la requête pour chaque requête exécutée. Les champs clés sont des identificateurs uniques des enregistrements dans la table de base de données de destination. L'adaptateur utilise ces valeurs lorsqu'il crée la requête pour l'exécution suivante. Cela lui permet de ne lire que de nouveaux enregistrements.

Par exemple, considérons l'instruction SQL suivante utilisée pour extraire des données de ticket d'incident.

```
Select ticket_id, status, organization, open_date, respond_date, resolved_date,
record_modified_date from t_ticket_data;
```

Dans cet exemple, le champ clé de la requête choisi pour obtenir les toutes dernières informations est record_modified_date et ce, afin de capturer tous les derniers enregistrements de la source de données. Il génère ainsi les nouveaux tickets ouverts depuis la dernière exécution de l'adaptateur, de même que les mises à jour de tickets existants. En sélectionnant ce champ en tant que champ clé de la requête, l'adaptateur ajoute automatiquement la section suivante à la fin de la requête pendant l'exécution :

```
where record_modified_date > : previous_value order by record_modified_date asc
```

Par conséquent, il ne récupère que les nouveaux enregistrements. Notez qu'un certain nombre de points sont à considérer lorsque vous choisissez le champ clé de requête, et ceux-là dépendent toujours du comportement de la source de données et de ce que vous voulez faire de ces données récupérées. Notez aussi que les champs choisis dans l'exemple précédent ne sont pas toujours le meilleur choix pour *toutes* les situations.

Un exemple de fichier DataSourceControl.xml pour un adaptateur SQL est présenté ci-dessous.

```
<AdapterControl Save="end" LastSaveTime="2005/05/20 15:59:02">
<Data><QueryCollection>

<Query QueryName="ChangeManagementOpenQuery">
<KeyField Name="Incident Ref"><LastValue>32357</LastValue></KeyField>
<KeyField Name="Date Logged"><LastValue>18/04/2005
16:56:26</LastValue></KeyField>
<LastFileName/>
```

```
</Query>

<Query QueryName="ChangeManagementPendingQuery">

<KeyField Name="Incident Ref"><LastValue>0</LastValue></KeyField>
<KeyField Name="Date Resolved"><LastValue>1900-01-
01</LastValue></KeyField><LastFileName/>

</Query>
```

send.txt

Tous les événements créés et prêts à être envoyés à CA Business Service Insight sont d'abord consignés dans le fichier à envoyer.

SendControl.xml

Le fichier sendcontrol.xml contient toutes les lignes qui ont été envoyées et ont été reconnues par CA.

Le fichier laisse l'adaptateur suivre le protocole d'acquittement de fenêtrage dynamique pour transférer les données à CA. Vous trouverez des informations supplémentaires sur ce mécanisme dans Communication de l'adaptateur (page 95).

```
<SendControl
PacketMaxSize="50"
LastAckSequence="47"
LastAckIndex="-1"/>
```

Fichier de sortie IllegalEvents (.txt)

L'adaptateur consigne dans le fichier de sortie IllegalEvents tous les enregistrements lus, mais pour lesquels une erreur s'est produite durant l'analyse. Cette erreur est généralement causée par la logique de validation introduite dans le fichier de configuration d'adaptateur. L'adaptateur enregistre ces enregistrements si le paramètre SaveIllegalEvents du fichier de configuration est défini sur Oui. Notez que le chemin d'accès de cette option doit également être défini en à l'aide de IllegalEventsDirectoryName. Ce dossier doit déjà exister, dans la mesure où il n'est pas créé automatiquement. Si le dossier n'existe pas, l'adaptateur envoie une erreur lors de son exécution.

Dans un adaptateur de fichier, le fichier contenant les enregistrements d'erreur est doté du même nom que le fichier source, tandis que dans l'adaptateur SQL, il porte le nom de la requête.

Translations.xml

Le fichier translation.xml contient les tables de procédé de conversion.

Lorsque l'adaptateur est exécuté en mode en ligne, le fichier contient une copie de la table de conversion de la base de données. Si la table de conversion est configurée comme étant distante, l'adaptateur charge la table de conversion de la base de données dans ce fichier et le remplace. S'il s'agit d'une configuration autonome, il lit le fichier local.

Lorsque l'adaptateur passe en mode hors ligne, il utilise le fichier uniquement comme table de conversion (pour plus d'informations sur les modes en ligne et hors ligne, consultez la section Modes d'exécution de l'adaptateur (page 101)).

```
<TranslationTableCollection>

<AssystResourceTable>
<Entry TranslationStatus="Ignored" resource="Authority"/>
<Entry TranslationStatus="Translated" resource="LONDON" TranslationTo="1006"/>

</AssystResourceTable>

<AssystSupplierEventTypeTable>
<Entry TranslationStatus="Translated" severity="1" TranslationTo="1545"/>
<Entry TranslationStatus="Translated" severity="2" TranslationTo="1550"/>
<Entry TranslationStatus="Translated" severity="3" TranslationTo="1551"/>

</AssystSupplierEventTypeTable></TranslationTableCollection>
```

Communication de l'adaptateur

Les adaptateurs interagissent avec la source de données d'une part et l'écouteur d'adaptateur de CA Business Service Insight et serveur de journalisation d'autre part, comme décrit dans le diagramme suivant.

L'adaptateur communique avec la source de données pour récupérer les données à l'aide d'une connexion ODBC et peut être situé localement ou à distance par rapport à la source de données, pourvu que l'adaptateur puisse établir la connexion ODBC.

L'adaptateur communique avec le serveur d'applications de CA Business Service Insight à l'aide du protocole TCP/IP et peut par conséquent être situé localement ou à distance par rapport à celui-là, pourvu qu'il puisse établir la connexion TCP/IP.

L'adaptateur doit avoir deux ports ouverts, un pour l'écouteur d'adaptateur et l'autre pour le serveur de journalisation. Les ports de l'écouteur d'adaptateur doivent être exclusifs à l'adaptateur et ne doivent pas entrer en conflit avec d'autres opérations ou applications de réseau qui pourraient également utiliser ces ports. Par exemple, veillez à ne pas utiliser le port 1521, dans la mesure où il est généralement utilisé par le protocole Oracle TNS pour communiquer avec la base de données, et ainsi de suite. Vous devrez peut-être également considérer tout pare-feu local qui peut bloquer ce trafic.

Remarque : Consultez votre administrateur local si vous ne savez pas quels ports sont disponibles, ou si vous devez demander l'ouverture de ports pour que la communication ait lieu.

Le port et l'adresse de l'écouteur d'adaptateur sont définis dans le fichier de configuration de l'adaptateur. Le port et l'adresse IP du serveur de journalisation sont définis via les entrées de l'adaptateur dans le registre.

L'option client/serveur de l'écouteur d'adaptateur est configurable, ce qui vous permet de configurer l'adaptateur en tant que client ou en tant que serveur. La configuration de l'opération client/serveur s'effectuée côté adaptateur dans les paramètres du fichier de configuration. Pour ce faire, les variables Port, Address et ConnectionInitiator doivent être définies en conséquence.

Si la variable ConnectionInitiator est définie sur l'adaptateur, seul un port de destination est exigé. Si elle est définie sur CA Business Service Insight, un port et une adresse IP de l'écouteur d'adaptateur de CA Business Service Insight sont requis. Par défaut, le serveur est défini sur l'adaptateur. Cette fonctionnalité peut parfois s'avérer importante pour permettre à une règle de pare-feu d'être déclenchée (une fonctionnalité dite de déclenchement de port). Parfois, un pare-feu n'autorise qu'une demande intérieure à un port, si un message a été envoyé depuis le pare-feu à ce même port. Le pare-feu autorise ainsi la communication.

Remarque : Consultez votre administrateur réseau pour plus d'informations sur les conditions locales qui peuvent affecter les communications de l'adaptateur.

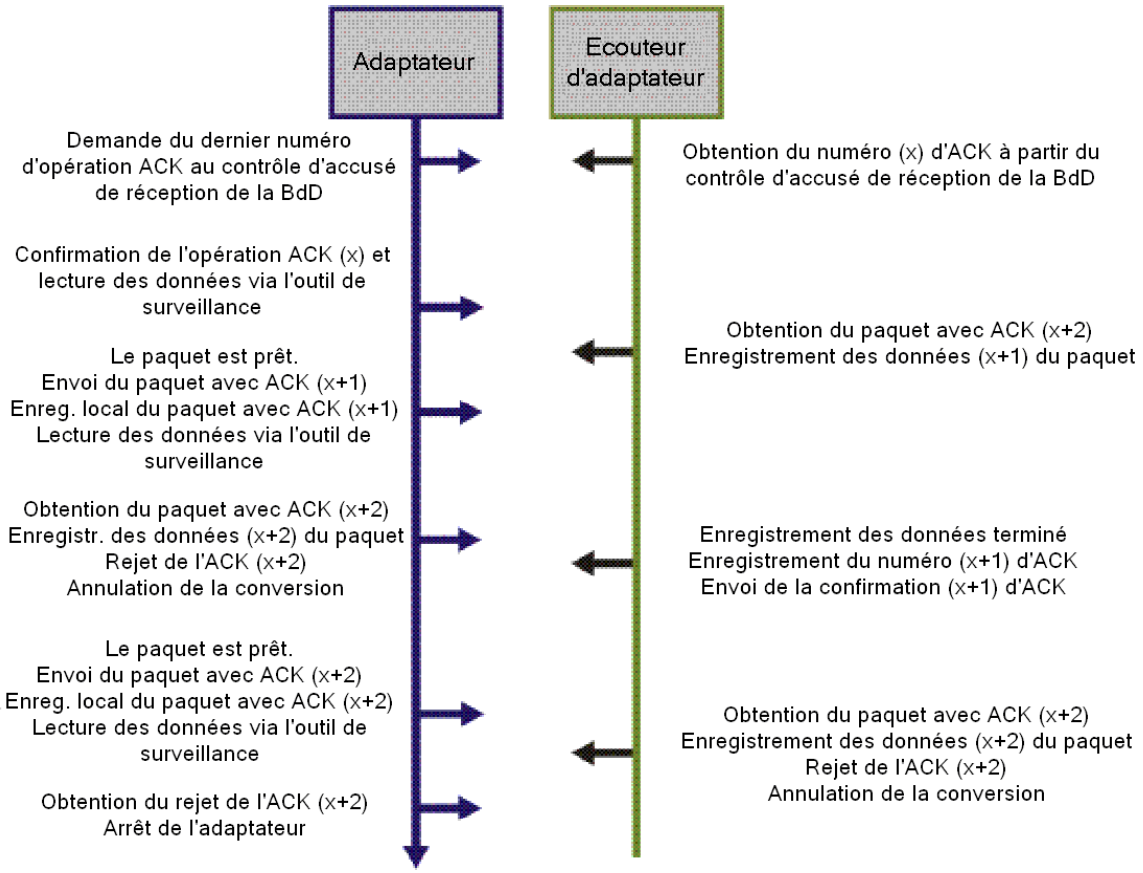
D'un point de vue de la sécurité, il est recommandé de définir l'adaptateur en tant que client, afin d'assurer la destination des événements lorsque vous travaillez dans un environnement de déploiement multiple pour l'analyse et la production.

Pour vérifier que la transmission des enregistrements de données entre l'adaptateur et l'écouteur d'adaptateur de CA Business Service Insight est réussie, l'adaptateur inclut un algorithme ACK/de fenêtrage dynamique qui se superpose à la couche TCP/IP. Cet algorithme envoie les données dans des paquets, puis attend l'acquittement de l'écouteur d'adaptateur avant de passer au paquet suivant. Chaque paquet contient plusieurs messages de données brutes. Vous pouvez configurer le nombre de messages dans un paquet en définissant le paramètre Taille du paquet. Chaque paquet a une séquence qui est contenue dans le message d'acquittement. Tous les paramètres pertinents qui contrôlent le processus sont contenus dans la section d'interface du fichier de configuration de CA Business Service Insight. Toutefois, vous n'avez généralement pas besoin de modifier ces paramètres.

L'écouteur d'adaptateur écrit les données brutes dans le paquet au cours d'une transaction unique.

Remarque : L'opération ACK ne s'applique qu'aux messages de données brutes envoyés à CA Business Service Insight.

L'illustration suivante présente le processus de communication de l'adaptateur.



Paramètres de registre de l'adaptateur

Dans les cas où la ligne de commande est incomplète, l'adaptateur utilise quelques-unes des définitions stockées dans le registre de système sur le serveur où l'adaptateur est installé.

Les entrées de registre sont écrites par le Gestionnaire d'adaptateurs, si cet utilitaire a été utilisé pour l'installation de l'adaptateur. Si ce n'est pas le cas, vous pouvez ajouter ces entrées manuellement dans le registre.

Remarque : Si vous installez un adaptateur dans un environnement Unix, ces entrées doivent être ajoutées manuellement puisqu'il n'existe aucun Gestionnaire d'adaptateurs pour cet environnement.

La liste ci-dessous répertorie les entrées de registre utilisées par l'adaptateur et l'utilitaire Gestionnaire d'adaptateurs.

Entrées de serveur générales

Les entrées suivantes sont écrites dans le registre de \HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters :

Propriétés possibles :

Nom	Type	Description
AdaptersDir	Chaîne	Répertoire racine pour tous les adaptateurs.*
FileAdapterConfTemplate	Chaîne	Chemin d'accès du modèle de configuration de l'adaptateur de fichier.* Le gestionnaire d'adaptateurs utilise ces informations pour copier le modèle de configuration dans un nouveau dossier de l'adaptateur à l'emplacement spécifié, dans le cadre du processus de création de l'adaptateur.
GenericFileAdapter	Chaîne	Exécutable de l'adaptateur de fichier.* Le gestionnaire d'adaptateurs peut créer un raccourci vers l'exécutable ou le copier dans un nouveau dossier de l'adaptateur à l'emplacement spécifié, dans le cadre du processus de création de l'adaptateur.
GenericSqlAdapter	Chaîne	Exécutable de l'adaptateur SQL.* Le gestionnaire d'adaptateurs peut créer un raccourci vers l'exécutable ou le copier dans un nouveau dossier de l'adaptateur à l'emplacement spécifié, dans le cadre du processus de création de l'adaptateur.
LogServerAddress	Chaîne	Adresse réseau du serveur de journalisation (Facultatif)** Port du serveur de journalisation : généralement 4040. (Facultatif)** Dans les cas où ces paramètres sont définis, l'adaptateur envoie un rapport concernant les messages de journal au serveur de journalisation de CA Business Service Insight.
LogServerPort	Chaîne	
SqlAdapterConfTemplate	Chaîne	Chemin d'accès du modèle de configuration de l'adaptateur SQL.* Le gestionnaire d'adaptateurs utilise ces informations pour copier le modèle de configuration dans un nouveau dossier de l'adaptateur à l'emplacement spécifié, dans le cadre du processus de création de l'adaptateur.

* Utilisé uniquement par l'utilitaire Gestionnaire d'adaptateurs

** Utilisé par l'adaptateur

Entrées d'adaptateur Individuelles

Les entrées suivantes sont écrites dans le registre de \HKEY_LOCAL_MACHINE\SOFTWARE\Oblicore\Adapters\<Adapter Name> :

Propriétés possibles :

Nom	Type	Description
ConfigurationFileName	Chaîne	Nom du fichier de configuration de l'adaptateur.**
Répertoire	Chaîne	Répertoire de l'adaptateur*
LogFileName	Chaîne	Nom du fichier de journal de l'adaptateur.**
Chemin d'accès	Chaîne	Chemin d'accès de l'exécutable de l'adaptateur*
RunAs	Nombre	Mode d'exécution.* Service/Application de console/Application Windows
Type	Nombre	Type d'adaptateur.* Fichier/SQL
LogFileMaxSize	Nombre	La valeur correspond à un nombre de Ko.** La plage autorisée est comprise entre 1 000 et 100 000 et la valeur par défaut est 1 000.

* Utilisé uniquement par l'utilitaire Gestionnaire d'adaptateurs

** Utilisé par l'adaptateur

Modes d'exécution de l'adaptateur

L'adaptateur peut être exécuté selon l'un des modes suivants :

Service **:**

Vous pouvez installer l'adaptateur comme vous le feriez pour un service Windows normal. Cela permet au système de contrôler sont statut (exécution, pause, arrêt, automatique) comme pour tout service Windows courant.

Pour installer l'adaptateur comme un service Windows, lancez le ficher exécutable de l'adaptateur à partir de la ligne de commande en utilisant **-i** pour l'installer comme service et **-u** pour le désinstaller.

Application **:**

Lancez le fichier exécutable de l'adaptateur à partir de la ligne de commande. Vous pouvez exécuter la ligne de commande de l'adaptateur de la manière suivante :

Options de ligne de commande :

```
TextFileAdapter.exe -i | -u | -v | -d [-t] [-f configurationFileName] [-l
logFileName] [-n serviceName]
[-a OblicoreAddress] [-p OblicorePort] [-la LogGerheaDed] [-lp LogServerPort]
```

Paramètre	Fonction
-i	Installer le service
-u	Supprimer le service
-v	Afficher la version
-d	Exécuter l'adaptateur en tant que console d'application
-t	Vérifier le fichier de configuration de contrôle seul et arrêter
-f	Définir le nom du fichier de configuration
-l	Définir le nom du fichier journal
-n	Définir le nom du service
-a	Définir l'adresse du serveur d'applications
-p	Définir le numéro de port du serveur d'applications (1024-49151)
-la	Définir l'adresse du serveur de journalisation
-lp	Définir le numéro de port du serveur de journalisation (1024-49151)

Ce type d'exécution est couramment utilisé dans les projets. Cela permet d'exécuter l'adaptateur via un fichier .bat**,** mais aussi d'utiliser le planificateur Windows pour contrôler le délai d'exécution de l'adaptateur. Pour planifier l'adaptateur à l'aide du planificateur Windows, il est nécessaire de configurer son mode d'exécution sur Une fois.

RunOnce : (facultatif [oui/non]). Lorsque cette option est définie dans le fichier de configuration sur Non, l'adaptateur fonctionne continuellement une fois exécuté. Lorsque cette option est définie sur Oui, l'adaptateur de fichier se lance, lit des enregistrements et s'arrête automatiquement lorsqu'aucun nouvel enregistrement ne s'affiche. Un adaptateur de fichier lit le fichier entier, puis attend quelques secondes et essaye de lire de nouveaux enregistrements (selon les paramètres SleepTime). S'il n'y a pas de nouveaux enregistrements, il s'arrête. Un adaptateur SQL exécute chaque requête une seule fois. Si RepeatUntilDry est défini sur Non, il s'arrête immédiatement. Si RepeatUntilDry est défini sur Oui, il attend (selon les paramètres SleepTime). Il essaye d'exécuter la requête à nouveau (selon le temps de veille de la requête), et s'il n'y a pas de nouveaux enregistrements, il s'arrête alors.

Pour plus d'informations sur les attributs SleepTime et RepeatUntilDry, reportez-vous à la section Spécifications de configuration de l'adaptateur (page 377).

La section relative à l'interface du fichier de configuration de CA Business Service Insight se compose d'attributs spécifiant les deux modes de connexion à CA Business Service Insight : en ligne et hors ligne.

Dans le mode en ligne, l'adaptateur se connecte à CA Business Service Insight, récupère les tables de conversion et les commandes de contrôle de CA Business Service Insight, puis renvoie des événements, des statuts et des demandes de conversion à CA Business Service Insight. Dans le mode hors ligne, l'adaptateur fonctionne avec un fichier de table de conversion local et écrit les événements dans un fichier de sortie.

Le mode hors ligne est communément utilisé lors du développement initial et du test d'un adaptateur.

Si ConsoleDebugMode est défini sur Oui, les messages de débogage s'affichent sur la console.

Pour plus d'informations sur les divers indicateurs, reportez-vous à la section Spécifications de configuration de l'adaptateur, (page 377) notamment à l'attribut ConsoleDebugMode.

Outils de configuration et de maintenance

Les outils nécessaires à la configuration des adaptateurs et à leur maintenance sont principalement des utilitaires autonomes, comme les utilitaires de CA Business Service Insight qui sont de simples fichiers exécutables Windows.

Remarque : Lors de la configuration des adaptateurs dans un environnement Unix, ces utilitaires ne sont pas disponibles, et la configuration doit être accomplie manuellement.

Lorsque vous travaillez sur un serveur d'applications de CA Business Service Insight, les utilitaires sont installés lors de l'installation de l'application et ils se trouvent dans le dossier Program Files\CA\Cloud Insight\Utilitaires.

Lors de l'installation, un raccourci est créé à partir du menu Démarrer. Il est recommandé d'utiliser ce raccourci pour exécuter les utilitaires.

Si l'un de ces utilitaires ne fonctionne pas sur le serveur d'applications de CA Business Service Insight, vous pouvez les copier sur un ordinateur Windows comme s'il s'agissait de fichiers standard, ou les installer à l'aide du package de CA Business Service Insight et en choisissant une installation personnalisée. La routine d'installation n'est toutefois pas obligatoire, et il suffit généralement de copier les fichiers .exe à un emplacement qui vous convient sur la station de travail. Cependant, si vous utilisez cette option, il se peut que vous n'ayez pas tous les fichiers .dll. Vous devez aussi les copier, si disponibles, depuis le serveur vers le dossier local sur la station de travail.

Configuration des adaptateurs et des conversions

Ce processus comprend les étapes suivantes :

1. Configuration de l'adaptateur à l'aide de l'assistant de configuration d'adaptateur ou en modifiant manuellement le fichier de configuration XML (décrit dans le chapitre suivant).
2. Déploiement de l'adaptateur.
3. Test de l'adaptateur.
4. Exécution de conversions.
5. Rédaction de scripts de conversion afin de prendre en charge un processus de conversion automatique (facultatif).

Remarque : Le déploiement de l'adaptateur peut s'effectuer automatiquement lorsque vous utilisez l'assistant de configuration d'adaptateur, dans la mesure où le nouveau service de déploiement d'adaptateur fonctionne en arrière-plan sur le serveur d'applications pour prendre en charge cette tâche.

Déployer un nouvel adaptateur (assistant de configuration d'adaptateur)

Lorsque vous créez un nouvel adaptateur à l'aide de l'assistant de configuration d'adaptateur, veillez à ce que les services « Ecouteur d'adaptateur » et les « Déploiement d'Adaptateur » soient en cours d'exécution.

Déployer un nouvel adaptateur (manuellement)

Conditions préalables à la création d'un nouvel adaptateur

Tous les éléments suivants doivent être en place avant de commencer à créer un nouvel adaptateur :

- **Dossier racine de l'adaptateur** : si CA Business Service Insight est installé sur le serveur, ce dossier existe sous le dossier racine Program Files\CA\Cloud Insight, dans le cas contraire, il doit alors être créé.
- **Dossier d'adaptateur individuel** : créez un dossier sous le dossier racine d'adaptateur pour l'adaptateur spécifique.

  Remarque : Si vous utilisez l'utilitaire Gestionnaire d'adaptateurs, il crée automatiquement le dossier lorsqu'il ajoute le nouvel adaptateur.
- **Fichiers exécutables de l'adaptateur** : TextFileAdapter.exe, l'exécutable de l'adaptateur de fichier texte pour l'adaptateur de fichier texte ; SQLAdapter.exe, l'exécutable de l'adaptateur de fichier SQL pour l'adaptateur SQL. Vous pouvez les trouver sur le serveur d'applications sous le dossier Program Files\CA\Cloud Insight\Adapters.

  Remarque : Pendant le processus de création de l'adaptateur via le Gestionnaire d'adaptateurs, vous devriez choisir, autant que possible, l'option de créer un raccourci vers les exécutables, plutôt que de créer une copie. Ainsi, lorsqu'une mise à niveau ou une Service Release est appliquée à CA Business Service Insight, tous les composants binaires sont mis à jour correctement.
- **Modèles de configuration** : modèles du fichier de configuration de l'adaptateur. Placez les fichiers dans le dossier racine de l'adaptateur. Vous les trouverez sur le serveur d'applications sous le dossier Program Files\CA\Cloud Insight\Adaptateurs. Les modèles de configuration sont utilisés pour créer le fichier de configuration sans devoir commencer à partir de zéro. Il est également courant d'utiliser un fichier de configuration d'adaptateur existant à cette fin.
- **Utilitaire Gestionnaire d'adaptateurs** : un exécutable autonome. Il suffit de faire une copie du fichier AdaptersManager.exe situé dans le dossier d'utilitaires sous le dossier Program Files\CA\Cloud Insight\Utilitaires du serveur d'applications. Il n'est pas nécessaire de créer l'adaptateur à l'aide de cet utilitaire, et vous ne pouvez utiliser cet utilitaire que sur un serveur Windows.
- **Deux ports TCP/IP ouverts** : un port pour l'écouteur d'adaptateur sur le serveur d'applications et un autre pour le serveur de journalisation. Le port du serveur de journalisation est habituellement 4040.

- Vérifiez quels composants de service d'application de CA Business Service Insight sont en cours d'exécution. Pour exécuter l'adaptateur, les composants de service suivants ont doivent être en cours d'exécution :
  - **Ecouteur d'adaptateur**
  - **Hôte de tâche**
  - **Journal de serveur**

Procédez comme suit :

1. Exécutez l'utilitaire Gestionnaire d'adaptateurs (voir la section relative au Gestionnaire d'adaptateurs ci-dessus).
2. Préparez toutes les conditions préalables mentionnées ci-dessus, puis créez un fichier séquentiel avec la ligne de commande exécutable (voir la section Modes d'exécution de l'adaptateur (page 101)).

Modifier le fichier de configuration de l'adaptateur

Lors de la création d'un adaptateur, le plus gros du travail consiste à modifier le fichier de configuration.

Ce travail implique de définir les attributs dans le fichier XML pour contrôler le comportement de l'adaptateur de sorte qu'il fonctionne comme prévu. Le fichier de configuration est un fichier XML dont chaque section correspond à une étape de son flux de travaux interne.

Les sections sont les suivantes :

- **Section Général** : attributs d'adaptateur variés, répertoire de travail, spécifications de fichiers de sortie, indicateur de débogage, valeurs par défaut, etc.

Oblicore :

- **Section OblicoreInterface** : attributs de la connexion avec le serveur CA Business Service Insight (port TCP/IP, mode de sécurité, etc.).
- **Section DatasourceInterface** : attributs de la connexion avec la source de données (chemin d'accès et modèle des fichiers, chaînes de connexion, requêtes SQL, etc.).
- **Section InputFormatCollection** : règles d'analyse pour analyser et manipuler le format de données d'origine (délimiteurs, types de champ, ordre des données, expressions régulières, etc.).
- **Section TranslatorCollection** : règles relatives à l'événement unifié de CA Business Service Insight, qui se compose de champs de données analysés et manipulés.
- **Section TranslationTableCollection** : règles de mappage de données entre la terminologie de données d'origine et les entités de données de CA Business Service Insight.

Ces sections sont décrites en détail à la rubrique Spécifications de configuration de l'adaptateur (page 377).

Remarque : L'ordre des noeuds XML dans chaque section n'est pas important.

Adaptateurs de fichier

Les adaptateurs de fichier utilisent le composant générique FileAdapter (exécutable) et le fichier de configuration requis pour l'analyse des fichiers ASCII.

Flux de travaux de l'adaptateur de fichier :

- Copiez le fichier source dans un fichier de travail et renommez-le.
- Lisez l'enregistrement logique.
- Analysez l'enregistrement en fonction des délimiteurs ou des expressions régulières.
- Recherchez la spécification correcte pour InputFormat.
- Créez l'enregistrement Evénement.
- Convertissez l'enregistrement Evénement.
- Mettez à jour le fichier de contrôle.

Exemple de fichier de configuration

L'exemple suivant utilise une source de données de fichier ASCII simple et sa configuration requise de sortie d'événement, et il présente les paramètres requis par le fichier de configuration de son adaptateur.

Vous pouvez afficher le fichier de configuration et le modifier en utilisant l'utilitaire XMLPad.

Pour une présentation de la structure et du contenu du fichier de configuration, consultez les sections pertinentes.

Remarque : Les paramètres présentés ne concernent que les paramètres d'attribut principaux et les plus importants. Pour obtenir les spécifications d'attribut complètes, reportez-vous à la section Spécifications de configuration de l'adaptateur (page 377).

Etude de cas d'un adaptateur de fichier

Considérons le système de surveillance de serveur suivant, qui produit des fichiers journaux avec des informations se conformant à la structure suivante :

Serveur	Horodatage	Utilisation de la mémoire	Utilisation de l'UC
Server001p	1.11.2004 0:00	26.71	78.85
Server002p	1.11.2004 1:00	49.09	89.49
Server001p	1.11.2004 2:00	51.37	57.51
Server001p	1.11.2004 3:00	23.53	98.9
Server002p	1.11.2004 4:00	86.21	85.11
Server002p	1.11.2004 5:00	39.98	35.97
Server001p	1.11.2004 6:00	70.58	55.96

Les fichiers sont placés dans le dossier d'adaptateur de CA Business Service Insight, dans un sous-dossier appelé Données. Les noms de fichier sont tous préfixés avec ServerData et suffixés avec la date et le temps. Les fichiers sont également des fichiers CSV avec l'extension .csv.

L'événement de sortie suivant est requis :

- **Champ de conversion de ressource** : serveur
- **Champ d'horodatage** : horodatage
- **Champs de données** : utilisation de mémoire, utilisation de l'UC

De plus, il est supposé que la source de données est située en Belgique (fuseau horaire CET avec +1 heure de décalage, auquel s'ajoute une heure supplémentaire pendant la période d'heure d'été).

Section générale du fichier de configuration

- **MajorVersion And MinorVersion** : à appliquer par défaut à la version d'application.
- **WorkingDirectoryName (facultatif)** : spécifie le chemin d'accès par défaut des fichiers de sortie d'adaptateur (contrôle de la source de données, conversion, envois). Dans ce cas, il est défini sur Sortie, et en tant que tel ce dossier est alors créé sous le dossier principal des adaptateurs.

Les indicateurs suivants contrôlent la façon dont l'adaptateur lit et convertit les enregistrements :

- **RunOnce** : s'il est défini sur Oui, l'adaptateur est lancé une fois et il lit toutes les données disponibles, puis s'arrête.
- **DataSourceDifferenceFromUTC** : indique le décalage horaire entre UTC (temps universel coordonné, sans jamais aucun décalage. Equivalent à GMT) et le fuseau horaire de la source de données. Le fuseau horaire de la source de données est le fuseau horaire avec lequel les champs d'heure de cette dernière sont indiqués. Cet attribut est nécessaire dans la mesure où l'adaptateur normalise toutes les dates selon le format UTC. Toutes les dates étant indiquées au format UTC, l'application a ainsi la flexibilité d'afficher l'heure en fonction des besoins de l'utilisateur. Les attributs suivants fournissent à l'adaptateur les détails concernant la manière de transformer les champs d'heure de l'entrée au format UTC :
  - **DefaultOffset** : décalage UTC en dehors de la période à l'heure d'été. Dans ce cas, il est défini sur « 1 » pour l'heure d'Europe centrale (CET).
  - **TimeFormat** : le format utilisé pour analyser les détails relatifs à l'heure d'été (décrite ci-dessous). Le format de temps européen est « jj/mm/aaaa hh:mi:ss », tandis que les spécifications de format de CA Business Service Insight sont définies comme « %d/%m/%Y %H:%M:%S ».
  - **DayLightSaving** : une période à l'heure d'été du fuseau horaire de la source de données. Cet élément est facultatif (autrement dit, s'il n'est pas sélectionné, l'heure d'été n'est pas appliquée) et il peut être choisi plus d'une fois ; une fois pour chaque période à l'heure d'été entrée. S'il existe plusieurs éléments, ils doivent être ordonnés chronologiquement, et les périodes ne doivent pas se chevaucher. Lors de la configuration des adaptateurs, cet élément est habituellement spécifié comme étant plusieurs années à l'avance. Dans ce cas, l'exemple de configuration correspond à une seule année.
- **De :** date de commencement de la période. Spécifiée en utilisant le format de temps défini au-delà du « 25/03/2007 02:00:00 ».

- **Pour** : date de fin de la période. Spécifiée en utilisant le format de temps défini au-delà du « 28/10/2007 03:00:00 ».
- **Décalage** : nombre d'heures de différence ajouté à l'attribut DefaultOffset au cours de la période à l'heure d'été. Saisissez 1, car il y aura une avance d'une heure pendant cette période à l'heure d'été spécifiée entre les 2 dates indiquées.

Section OblicoreInterface du fichier de configuration

Cette section définit la connexion au serveur CA Business Service Insight.

- Mode : dans le mode en ligne, l'adaptateur se connecte à CA Business Service Insight, récupère les tables de conversion et la commande de contrôle de CA Business Service Insight, puis renvoie des événements, des statuts et des demandes de conversion à CA Business Service Insight. Configurez l'adaptateur dans ce mode et définissez la valeur sur En ligne.
- Port : le numéro de port TCP/IP utilisé par l'adaptateur pour communiquer avec le serveur CA Business Service Insight (à l'emplacement de l'écouteur d'adaptateur).

  En supposant que ceci ne pose pas de problème, configurez cet adaptateur pour utiliser le port « 5555 » (choisi arbitrairement). Ce choix doit aussi être spécifié sur le serveur dans l'interface utilisateur graphique de l'adaptateur et ce, afin d'activer la communication.

Section DataSourceInterface du fichier de configuration

La section Interface de source de données est constituée d'attributs spécifiant la connexion et le type de connexion entre l'adaptateur et la source de données. Il existe deux types d'interface, fichier et SQL. La différence principale entre les deux est que pour les fichiers, l'ensemble des fichiers est exigé, tandis que pour SQL, l'ensemble des requêtes est exigé.

La section Interface de source de données définit également la façon dont l'adaptateur gère le fichier source (s'il supprime le fichier d'origine s'il a été créé uniquement pour l'adaptateur, ou s'il laisse les données au cas où elles pourraient avoir d'autres utilisations, et ainsi de suite).

Dans le cas des adaptateurs de fichier, pour lire et analyser des fichiers ASCII, l'interface de fichiers est utilisée comme dans l'illustration suivante. Choisissez les valeurs suivantes pour les paramètres comme suit :

La section Fichiers sous le noeud Interface de source de données se rapporte à la connexion à la source de données. Configurez les attributs suivants.

Remarque : Cette section est complètement différente pour un adaptateur SQL.

- **DeleteFileAfterProcessing** : définit la façon dont l'adaptateur gère le fichier source et détermine comment l'adaptateur contrôle la lecture afin de ne lire que de nouveaux enregistrements. Dans ce cas, les fichiers sources restent à leur place sur le serveur et cette valeur est définie sur Non.

  Si un fichier est créé uniquement pour l'adaptateur et qu'il puisse être supprimé après son cours de traitement, l'attribut doit être défini sur Oui. Le fichier est alors renommé, traité et supprimé.

  Lorsque l'attribut est défini sur Non, le fichier est copié et le traitement a lieu en utilisant le fichier copié. Si de nouveaux enregistrements sont ajoutés à la fin de ce fichier, l'adaptateur copie ces nouveaux enregistrements dans le fichier de travail pendant le cycle suivant. Si de nouveaux enregistrements ne sont pas ajoutés au fichier, l'adaptateur cherche le fichier suivant avec les mêmes modèle et nom (dans l'ordre lexicographique) que le fichier actuel. Si l'adaptateur trouve un tel fichier, il procède à son cours de traitement. L'adaptateur ne revient pas au fichier précédent même si de nouveaux enregistrements ont été ajoutés.

  Définissez l'attribut sur Non lorsque vous avez besoin de garder l'intégrité du fichier source et lorsque le fichier doit être ajouté.

- **InputFormat** : se réfère au nom donné à l'élément InputFormat dans l'ensemble InputFormatCollection suivant qui prend en charge les données de cette connexion de source de données. Le format d'entrée correspond à la structure des champs des données entrées en provenance de la source de données, une fois qu'elles ont été analysées par l'adaptateur. La méthode d'analyse est spécifiée dans l'attribut des délimiteurs comme expliqué ci-dessous. Lorsque vous utilisez plus d'une connexion avec des formats d'interface différents, ce champ revêt une plus grande importance et il détermine quelle structure de format d'entrée manipule chaque donnée des sources de données.
- **Path** : emplacement physique des fichiers de source de données. Par exemple : C:\Program Files\CA\Cloud Insight\Adapters\ServersAdapter\data\.
- **NamePattern** : spécifie le nom du fichier de source de données. Les caractères génériques peuvent être utilisés si plus d'un fichier utilise le même format d'entrée. Si un nom de fichier est spécifié sans caractères génériques, l'adaptateur cherche uniquement un fichier avec ce nom. Si des caractères génériques sont utilisés, l'adaptateur recherche tous les fichiers qui correspondent au modèle, les trie lexicographiquement, puis les lit un par un. Pendant l'exécution suivante, il cherche de nouveaux enregistrements dans le dernier fichier avant de passer au suivant.

  Dans cet exemple, si le caractère générique « * » est utilisé, alors les valeurs d'attribut est « ServerData*.csv ». (L'adaptateur lit tous les fichiers dont les noms commencent par ServerData et portent l'extension .csv.)

  Important : Il est recommandé d'ajouter une date et une heure à la fin des noms de fichier utilisant le format AAAAMMJJ-HHMISS suivant, afin de s'assurer que les fichiers sont triés correctement, qu'ils sont lus dans l'ordre voulu et qu'aucun ne soit ignoré. Vous pouvez également ajouter la portion d'heure si plusieurs fichiers sont produits chaque jour.
- **Delimiters** : Méthode de détermination du mode d'analyse du fichier. Vous pouvez spécifier un ou plusieurs caractères comme délimiteurs, en fonction des lignes de données à analyser en champs. La valeur par défaut est "/t".

Le fichier de source de données de cet exemple est un fichier CSV. La façon la plus simple de faire l'analyse de ces fichiers est de spécifier la virgule comme délimiteur.

D'autres méthodes d'analyse sont disponibles :

- **RegExForParser** : Utilise une expression régulière pour définir la règle d'analyse grammaticale.

- **LogicLineDefinition** : Utilisée lorsqu'une ligne du fichier est créée à partir de plusieurs lignes.
- **TitleExists** (facultatif) : Spécifie si la première ligne non vide dans le fichier de source de données est une ligne de titre. Le titre peut être utilisé par l'adaptateur lors de l'analyse des données. Dans cet exemple, chaque fichier de données contient la ligne de titre, donc « oui » doit être spécifié pour cet attribut.

Section InputFormatCollection du fichier de configuration

Cette section spécifie la structure de données récupérée depuis la source de données, la façon dont une ligne de données doit être découpée en champs ainsi que les types de ces champs et leurs formats. Le filtrage des données initiales et les manipulations de données peuvent être effectués dans cette section en utilisant les champs InputFormatSwitch et Compound respectivement.

Dans cette section, il est possible de définir la métrique de validation pour les enregistrements d'entrée à l'aide de InputFormatValidation et de ValidationCase, qui déterminent si un enregistrement est légal.

Le noeud InputFormatCollection peut contenir un ou plusieurs noeuds InputFormat.

Le flux de travaux général suivi par l'adaptateur durant cette section est le suivant :

- La ligne de données est comparée à l'InputFormat spécifié dans DataSourceInterface.
- Les données sont réparties dans des champs en fonction de la spécification InputFormat correspondante. L'ordre de InputFormatFields doit correspondre à celui des champs analysés à partir de la source de données.
- Les valeurs sont affectées aux champs composés en combinant et en séparant les champs de données spécifiés précédemment. Les définitions des champs composés doivent suivre celles des champs normaux.
- Les données traitées sont comparées aux conditions de TranslatorSwitch afin de déterminer le convertisseur utilisé pour créer l'événement de sortie à partir de l'enregistrement d'entrée.

- Les données traitées sont envoyées au convertisseur correspondant ou ignorées.

Utilisez les paramètres suivants :

- **InputFormatName** : Nom de ce format, référencé par la section DatasourceInterface.
- **InputFormatFields** : InputFormatFields peut contenir un ou plusieurs noeuds de champ, en fonction du nombre de sources de données des champs de saisie.
- **InputFormatField** : Spécifie un champ de données de la ligne de données d'origine, ou un champ composé.

  <InputFormatField Name="timestamp" Type="time"

  TimeFormat="%d/%m/%Y %H:%M:%S"/>

  - **Name** : Nom de ce champ, qui sera référencé comme champ source par les autres éléments, comme l'élément composé ou les champs de TranslatorFields.
  - **Type** : Type de données du champ (string/integer/real/time).
  - **Source** : Valeur de la source pour ce champ. Valeur par défaut : « événement » ; valeurs possibles :
    - **event** : la valeur du champ est fournie par l'événement venant de la source de données. Les valeurs des champs sont prises dans l'ordre dans lequel elles proviennent de la source de données.
    - **compound** : la valeur du champ est créée à partir d'autres champs, selon une manipulation des valeurs des autres champs ou de constantes. Les champs manipulés doivent déjà être définis.
    - **title** : la valeur du champ provient des noms des champs de titre. Le champ référencé doit déjà être défini.
    - **filename** : la valeur du champ provient du nom du fichier de source de données ; uniquement pour les adaptateurs de fichiers texte.
    - **constant** : la valeur du champ est une constante provenant de la valeur de ConstantValue dont la propriété doit apparaître ensuite.
- **TranslatorSwitch** : Détermine le convertisseur à utiliser pour traduire la ligne de données en un événement unifié.
  - **DefaultTranslator** : Convertisseur à utiliser lorsque les critères ne correspondent pas. Si la valeur est définie comme « Ignorer », aucun convertisseur n'est utilisé et la ligne est ignorée et non transmise à CA Business Service Insight.

Section TranslatorCollection du fichier de configuration

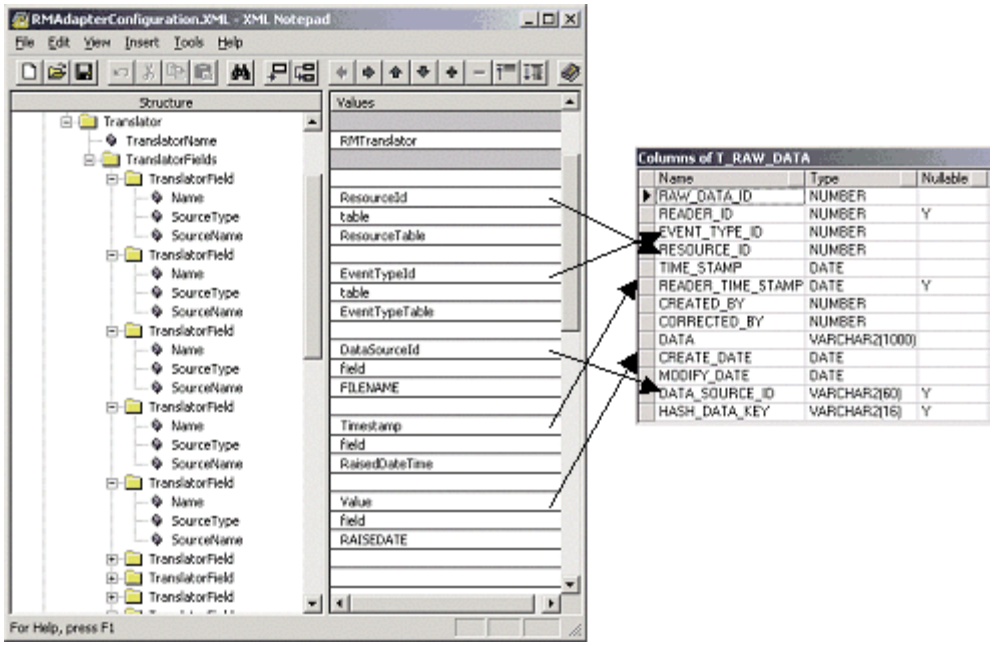
La section Translator Collection définit comment l'enregistrement de source de données analysé et manipulé extrait des sections précédentes sera converti en un événement CA Business Service Insight.

Cette section définit également comment traiter des événements dupliqués et comment utiliser le mécanisme de singularité d'événement (pour plus de détails, voir Singularité d'événement (page 147)).

Lorsque le mode d'interface est défini comme en ligne, l'événement CA Business Service Insight a une structure unifiée qui contient les champs suivants :

- **Timestamp** : Moment d'occurrence de l'événement.
- **ResourceId** :ID de ressource associée à l'événement (la ressource mesurée dans cet événement).
- **EventTypeId** : Type d'événement associé à l'événement ; décrit le type d'événement (type de dimension de la ressource, type de l'action de ticket, etc.).
- **DataSourceId** (facultatif) : Toute valeur. Fournit des critères supplémentaires de filtrage pour les événements de données brutes.
- **Value** (multiple) : Valeur(s) de l'événement (résultat de dimension, numéro de ticket, etc.). Ce champ s'affiche souvent plusieurs fois.

La structure du convertisseur correspond à la structure du type d'événement dans CA Business Service Insight, et également à la table de base de données T_RAW_DATA qui contient l'événement, comme illustré ci-dessous :



- **Translator :** décrit la méthode de conversion de l'ensemble de champs qu'il reçoit dans un événement de sortie.
- **TranslatorName** : Nom utilisé par InputFormat pour envoyer les définitions de champ à ce convertisseur. Cet exemple utilise le nom « Translator1 ». L'illustration précédente contient des valeurs utilisables pour ce scénario.
- **TranslatorFields** : Contient une liste d'éléments TranslatorField, dont chacun contient les attributs suivants :
  - **Name** : Nom de champ. Dans l'interface en ligne ce nom doit être Horodatage, ResourceId, EventTypeId, ResourceId ou Valeur.
  - **SourceType** : Spécifie la source de la valeur du champ. Elle peut être l'un des éléments suivants :
    - **Field** : La valeur de ce champ provient du champ au format d'entrée. L'attribut SourceName contient le nom de champ d'InputFormat.
    - **Table** : La valeur de ce champ provient de la table de conversion. L'attribut SourceName contient le nom de la table de conversion qui référence un nom défini dans la section suivante de TranslationTableCollection. Ce type est utilisé pour les valeurs qui sont choisies pour être converties en ressources et types d'événement dans l'événement.

- **Lookup** : La valeur de ce champ provient de la table de conversion. L'attribut SourceName contient le nom de la table. Valeur à convertir de l'attribut LookupValue et non du format d'entrée. Elle est couramment utilisée lorsqu'une valeur constante est nécessaire pour les conversions.
- **Constant** : La valeur de ce champ est une constante provenant de l'attribut ConstantValue. Lorsqu'une valeur constante est utilisée, il est nécessaire de spécifier le type de champ à l'aide de l'attribut Type.

- **SourceName** : Contient le nom du champ ou de la table de conversion.
- **LookupValue** : Contient la valeur de l'élément de recherche lorsque SourceType="lookup".
- **ConstantValue** : Contient la valeur constante lorsque SourceType=constant. Dans le cas d'un champ horaire, la valeur constante est formatée selon le format de temps TimeFormat (défini dans la section Général de l'adaptateur) ou « Now » ou « NowUtc », où « Now » correspond à l'heure actuelle dans le paramétrage régional de la source de données et « NowUtc » correspond à l'heure en temps universel.

Cette section contient également les tables de mappage définissant le mappage des valeurs de source de données en champs événements de CA Business Service Insight et contient la définition de table avec la valeur de source de données référencée à convertir.

- **LoadingMode** : La valeur par défaut est « distante » pour l'interface en ligne et « autonome » pour l'interface hors ligne.

  La méthode de chargement des tables de conversion doit être spécifiée comme suit :

  - **Standalone** : L'adaptateur charge les tables de conversion localement. Il n'y a aucune connexion au serveur CA Business Service Insight pour la conversion. Les changements dans les tables de conversion sont stockés uniquement dans le fichier local.
  - **Remote** : L'adaptateur envoie une demande pour charger toutes les tables du serveur CA Business Service Insight. Les changements apportés dans les tables de conversion sont également stockés localement.
- **TranslationTable** : Lie la valeur d'événement à la table de mappage.
  - **Name** : Nom de la table de conversion utilisée et référencée par le convertisseur.

- **DestinationType** : [ressource, event_type, contract_party, service, time_zone, value]. Contient le type de table de conversion. Dans cet exemple, la colonne **Serveurs** du fichier de source de données est convertie en ressource CA Business Service Insight. Par conséquent, le type de la table de conversion est Ressource, et elle contient les conversions des valeurs des ID Ressource dans CA Business Service Insight.
- **TranslationField** : Nom du champ à convertir, provenant des champs de format d'entrée. Peut contenir un maximum de cinq champs.

Chaque table de conversion définie dans le fichier de configuration doit correspondre à une définition dans l'interface utilisateur de CA Business Service Insight.

La représentation XMLE d'un fichier de configuration échantillon est comme suit :

```
<?xml version="1.0" encoding="utf-8"?>
<AdapterConfiguration>
     <General MajorVersion="4" MinorVersion="0" RunOnce="yes" LogDebugMode="yes"
ConsoleDebugMode="yes" WorkingDirectoryName="output"
RejectedEventsUpperLimit="10000">
     <DataSourceDifferenceFromUTC DefaultOffset="0" TimeFormat="%d/%m/%Y %H:%M">
          <DaylightSaving From="20/04/2001 00:00" To="15/10/2001 00:00"
Shift="1"/>
     </DataSourceDifferenceFromUTC>
</General>
     <OblicoreInterface Mode="online">
          <OnlineInterface Port="5555" SecurityLevel="none"/>
     </OblicoreInterface>
     <DataSourceInterface>
          <Files>
               <File DeleteFileAfterProcessing="no" InputFormat="InputFormat1"
NamePattern="servers*.csv" Path=" C:\Program
Files\Oblicore\Adapters\ServersAdapter\data\" TitleExists="yes" SleepTime="60"
Delimiters=","/>
      </Files>
</DataSourceInterface>
<InputFormatCollection>
     <InputFormat InputFormatName="InputFormat1">
          <InputFormatFields>
               <InputFormatField Name="resource" Type="string"/>
                    <InputFormatField Name="timestamp" Type="time"
TimeFormat="%d.%m.%Y %H:%M"/>
                         <InputFormatField Name="memory_util" Type="real"/>
                         <InputFormatField Name="cpu_util" Type="real"/>
                </InputFormatFields>
     <TranslatorSwitch DefaultTranslator="Translator1"/>
```

```
     </InputFormat>
</InputFormatCollection>
<TranslatorCollection>
<Translator TranslatorName="Translator">
         <TranslatorFields>
            <TranslatorField Name="ResourceId" SourceType="table"
SourceName="ResourceTable"/>
            <TranslatorField Name="EventTypeId" SourceType="lookup"
SourceName="EventTable" LookupValue="PerformanceEvent"/>
            <TranslatorField Name="Timestamp" SourceType="field"
SourceName="timestamp"/>
            <TranslatorField Name="Value" SourceType="field"
SourceName="memory_util"/>
            <TranslatorField Name="Value" SourceType="field"
SourceName="cpu_util"/>
         </TranslatorFields>
      </Translator>
   </TranslatorCollection>
   <TranslationTableCollection LoadingMode="remote">
      <TranslationTable Name="ResourceTable" DestinationType="resource">
         <TranslationField>resource</TranslationField>
      </TranslationTable>
      <TranslationTable Name="EventTable" DestinationType="event_type">
         <TranslationField>resource</TranslationField>
      </TranslationTable>
   </TranslationTableCollection>
</AdapterConfiguration>
```

Adaptateurs SQL

Les adaptateurs SQL représentent pour l'essentiel l'utilisation du composant générique adaptateur SQL (exécutable SQL Adapter) avec les paramètres appropriés dans le fichier de configuration. L'adaptateur SQL peut se connecter à toutes les sources de données qui prennent en charge ODBC et OLEDB. La connexion est établie via une chaîne de connexion. Le pilote approprié doit être installé sur le serveur sur lequel l'adaptateur est installé.

Exemples de sources de données :

- Oracle
- SQL Server
- Accès
- Excel
- Fichiers texte, fichiers CSV (il est également possible de se connecter à ces derniers à l'aide de l'adaptateur TEXT, mais la connexion ODBC offre souvent des fonctionnalités de filtrage et de requête supplémentaires).

Flux de travaux de l'adaptateur SQL :

- Ouvrez la connexion.
- Remplacez les variables locales par les dernières valeurs des champs clés.
- Utilisez le remplissage automatique, créez les clauses WHERE des requêtes et concaténez jusqu'à la fin des requêtes.

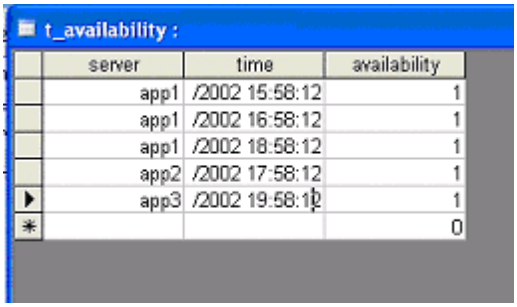
  Exécutez la requête et recevez le jeu d'enregistrements.

  Pour chaque enregistrement du jeu d'enregistrements renvoyé par la requête :

  - Recherchez la valeur de InputFormat correcte.
  - Créez l'enregistrement Evénement.
  - Convertissez l'enregistrement.
  - Mettez à jour la valeur des champs clés dans le fichier de contrôle.
- Fermez la connexion.
- Passez en mode veille.

Exemple de fichier de configuration de l'adaptateur SQL

Soit une base de données MS Access (.mdb) contenant la table suivante :



t_availability :

server	time	availability
app1	/2002 15:58:12	1
app1	/2002 16:58:12	1
app1	/2002 18:58:12	1
app2	/2002 17:58:12	1
app3	/2002 19:58:12	1
		0

La seule différence entre les fichiers de configuration de l'adaptateur SQL et de l'adaptateur de fichier réside dans la section DatasourceInterface.

La section DatasourceInterface d'un adaptateur de fichier stocke la collection Files, dans laquelle le fichier de l'adaptateur SQL contient ConnectionString et QueryCollection.

La différence principale entre les deux fichiers de configuration réside dans la méthode d'extraction et d'analyse, le reste des fichiers est identique.

L'interface SQL définit la connexion à la base de données et les requêtes utilisées pour récupérer les données.

Les détails sont les suivants :

Remarque : La section est définie d'après la base de données de la source de données susmentionnée.

Elément de chaîne de connexion

ConnectionString : Chaîne de connexion pour l'accès à la source de la base de données.

Vous pouvez définir ConnectionString dans l'élément DataSourceInterface et/ou dans les éléments de la requête. La définition de ConnectionString dans l'élément DataSourceInterface est la définition par défaut, et elle est utilisée uniquement dans une requête sans définition de ConnectionString. Cela permet à l'adaptateur se connecter à plusieurs bases de données dans lesquelles chaque requête peut avoir sa propre chaîne de connexion. Pour plus de détails sur le mécanisme de chaîne de connexion, consultez la section suivante.

Section Ensemble de requêtes du fichier de configuration

Query : Spécifie les attributs de la requête.

- **QueryName** : Nom de la requête.
- **InputFormat** : InputFormat associé à cette requête. L'adaptateur utilise InputFormat pour extraire les données de la source.

  Remarque : L'ordre des champs InputFormat doit correspondre à celui des champs sélectionnés dans la requête.
- **SleepTime** : Durée (en secondes) pendant laquelle l'adaptateur attend l'arrivée de nouvelles données.
- **SelectStatement** : Contient l'instruction sélectionnée à exécuter sur la base de données source.

  Remarque : Vous devez entrer les champs clés de la requête comme premières valeurs sélectionnées dans l'instruction.

  - **AutoCompleteQuery** : Lorsque cet attribut est défini sur oui, l'adaptateur concatène automatiquement une instruction WHERE pour la requête spécifiée, comme suit :
    - Création d'une instruction WHERE qui récupère uniquement les nouvelles valeurs à partir des champs clés.
    - Tri des résultats en fonction des champs clés.
- **QueryKeyFields** : Définition des champs où commencer l'extraction de données suivante.

- **KeyField** :
  - **Name** : Nom du champ, provenant des champs de la requête.
  - **Sort** : Ordre de tri des données (croissant/décroissant).
  - **ValueLeftWrapper** : Concatène les caractères avant la valeur du champ. La valeur par défaut est ' (apostrophe).
  - **ValueRightWrapper** : Concatène les caractères après la valeur du champ. La valeur par défaut est ' (apostrophe).
  - **Critical** : Interrompt l'exécution des autres requêtes si celle-ci échoue.
  - **SleepTime** : Temps d'attente entre les opérations nécessaires. La valeur par défaut est ' (apostrophe).

  Remarque : Les champs Date nécessitent souvent d'utiliser des caractères spéciaux pour entourer la date elle-même. Les caractères nécessaires pour identifier le champ comme champ de date dépendent de la source de données. Vous pouvez utiliser le caractère #, comme dans l'illustration en début de cette section, pour encadrer le champ de cette valeur dans Excel. Toutefois, d'autres sources de données peuvent nécessiter des méthodes différentes. Pour plus informations, consultez la section Spécifications de configuration de l'adaptateur (page 377).

- **SelectInitialValues** : Instruction SELECT qui fournit les valeurs initiales aux champs clés pour la première instruction WHERE (lorsque le fichier de contrôle est vide).

  Exemple de requête (ODBC basé sur Excel) créée avec AutoCompleteQuery="yes" :

  SELECT INC_CLOSE_DATE, INCIDENT_REF, Severity, `Resolve Time`, `Date Logged`, `Date Resolved` FROM `AllCalls$` WHERE INC_CLOSE_DATE>CDate('7/03/2005 13:06:21') order by INC_CLOSE_DATE

  Cette instruction SELECT doit pouvoir être exécutée sur la base de données cible vers laquelle la requête s'exécute. Elle peut différer en fonction des sources et des pilotes ODBC utilisés pour s'y connecter. Par exemple, dans Oracle il est possible de sélectionner les valeurs dans la table spéciale "dual" (select 'aaa', '1-jan-1970' from dual), mais dans Excel vous pouvez simplement sélectionner les valeurs directement, sans table. (select 'aaa')

Voici un fichier de configuration complet en XML :

```
<?xml version="1.0" encoding="utf-8"?>
<AdapterConfiguration>
  <General MajorVersion="3" MinorVersion="0" RunOnce="yes" LogDebugMode="yes"
ConsoleDebugMode="yes" WorkingDirectoryName="d:\Oblicore\Training
Kit\Exercises\Adapters\SQL Adapters\Ex1\Solution">
```

```
      <DataSourceDifferenceFromUTC DefaultOffset="1" TimeFormat="%Y/%m/%d-
%H:%M:%S"/>
   </General>
   <OblicoreInterface Mode="online">
      <OnlineInterface Port="2000" SecurityLevel="none"/>
   </OblicoreInterface>
   <DataSourceInterface>
      <ConnectionString>

          Driver={Microsoft Access Driver (*.mdb)};Dbq=d:\Oblicore\Training
Kit\Exercises\Adapters\SQL Adapters\Ex1\db1.mdb;

      </ConnectionString>
      <QueryCollection>
         <Query QueryName="Query" InputFormat="InputFormat" SleepTime="5">
            <SelectStatement AutoCompleteQuery="yes">

               select time,server,availability from t_availability

          </SelectStatement>
            <QueryKeyFields>
               <KeyField Name="time" Sort="asc" ValueLeftWrapper="#"
ValueRightWrapper="#"/>
               <KeyField Name="server" Sort="asc"/>
               <SelectInitialValues>

                     select 'AAA','1/1/1970'

               </SelectInitialValues>
            </QueryKeyFields>
         </Query>
      </QueryCollection>
   </DataSourceInterface>
   <InputFormatCollection>
      <InputFormat InputFormatName="InputFormat">
         <InputFormatFields>
            <InputFormatField Name="timestamp" Type="time" TimeFormat="%m/%d/%Y
%I:%M:%S %p"/>
            <InputFormatField Name="server" Type="string"/>
            <InputFormatField Name="status" Type="integer"/>
         </InputFormatFields>
         <TranslatorSwitch DefaultTranslator="Translator"/>
      </InputFormat>
   </InputFormatCollection>
   <TranslatorCollection>
      <Translator TranslatorName="Translator">
         <TranslatorFields>
            <TranslatorField Name="ResourceId" SourceType="table"
SourceName="ResourceTable"/>
```

```
            <TranslatorField Name="EventTypeId" SourceType="constant"
ConstantValue="1500"/>
            <TranslatorField Name="Timestamp" SourceType="field"
SourceName="timestamp"/>
            <TranslatorField Name="Value" SourceType="field"
SourceName="status"/>
         </TranslatorFields>
      </Translator>
   </TranslatorCollection>
   <TranslationTableCollection LoadingMode="remote">
      <TranslationTable Name="ResourceTable" DestinationType="resource">
         <TranslationField>server</TranslationField>
      </TranslationTable>
   </TranslationTableCollection>
</AdapterConfiguration>
```

Chaîne de connexion de l'adaptateur SQL

Le mécanisme de chaîne de connexion de l'adaptateur SQL est conçu pour fournir les fonctionnalités suivantes :

- Utiliser plusieurs chaînes de connexion dans le même adaptateur.
- Utiliser un modèle de chaîne de connexion, dans lequel il est possible de modifier le nom de fichier sans changer de fichier de configuration.
- Supprimer la source de données du fichier lorsque l'adaptateur a fini de le lire.
- Déplacer la source de données du fichier vers un autre emplacement lorsque l'adaptateur a fini de le lire.

Outre la définition simple de la chaîne de connexion sous forme de chaîne dans l'élément ConnectionString, la chaîne de connexion peut être définie par des segments, chaque segment contenant les valeurs spécifiques qui sont concaténées vers la chaîne de connexion. L'adaptateur peut ainsi créer la chaîne de connexion dynamiquement.

Il existe deux types de segment. L'un est constitué de texte et contient le texte concaténé vers la chaîne de connexion. Le deuxième est un fichier qui contient un nom de fichier avec ou sans caractères génériques. Le segment de type fichier ne peut apparaître qu'une fois. Il contient d'autres attributs qui définissent ce qui doit être fait avec le fichier lu.

Vous pouvez définir ConnectionString dans l'élément QueryCollection et/ou dans les éléments de la requête. La définition de ConnectionString dans l'élément QueryCollection est la définition par défaut, et elle est utilisée uniquement dans une requête sans définition explicite de ConnectionString.

Eléments et attributs

Elément ConnectionString

Cet élément peut contenir les éléments enfants du segment. S'il contient au moins un élément de segment, la chaîne de connexion est concaténée avec celui-ci. Dans le cas contraire, la chaîne de connexion est prélevée du texte de l'élément ConnectionString (comme dans la version actuelle).

Elément Segment

Cet élément contient un attribut obligatoire appelé *Type*, qui a deux valeurs possibles, "text" et "file". Lorsque le type est file, l'élément Segment doit contenir au moins un élément File. Chaque élément File est considéré comme une requête différente.

Elément File

Cet élément contient des attributs qui définissent les fichiers à utiliser dans la chaîne de connexion et ce qui doit être fait du fichier lorsque l'adaptateur a fini de le lire.

- **Path** : Définit le chemin d'accès au fichier (répertoire).
- **NamePattern** : Définit le nom de fichier avec le chemin d'accès indiqué. La valeur peut contenir des caractères génériques.
- **UsePath** : Valeurs possibles : yes/no. La valeur par défaut est yes. Si elle est définie sur yes, le chemin d'accès au fichier est concaténé à la chaîne de connexion.
- **UseFileName** : Valeurs possibles : yes/no. La valeur par défaut est yes. Si elle est définie sur yes, le nom de fichier est concaténé à la chaîne de connexion (nécessaire pour les fichiers Excel).
- **UsePath** : Valeurs possibles : yes/no. La valeur par défaut est no. Si elle est définie sur yes, l'adaptateur met à jour le fichier schema.ini avec le nom de fichier actuel.

  Remarque : Utilisez cet attribut uniquement lorsque vous voulez que l'adaptateur modifie le fichier schema.ini.

Variables internes

Deux variables internes supplémentaires peuvent être utilisées dans les éléments SelectStatement et SelectInitialValues, à savoir :

- :file_name : remplacée par le nom de fichier actuel, extension comprise.

- *:file_name_no_ext* : remplacée par le nom de fichier actuel sans l'extension.

Exemples

Exemple 1 : Chaîne de connexion simple pour Oracle :

```
<ConnectionString> Provider=msdasql;dsn=0 ; uid=0 ; pwd=0 </ConnectionString>
```

ou

```
<ConnectionString>
   <Segement Type="text" Text="Provider=msdasql;"/>
   <Segement Type="text" Text="dsn=0; "/>
   <Segement Type="text" Text="uid=0; "/>
   <Segement Type="text" Text="pwd=0; "/>
</ConnectionString>
```

Exemple 2 : Chaîne de connexion simple vers Excel avec un fichier unique :

```
<ConnectionString>Driver={Microsoft Excel Driver (*.xls)}; DriverId=790;
                                Dbq=d:\Oblicore\Availabilty_2003_01.XLS
</ConnectionString>
```

ou

```
<ConnectionString>
   <Segement Type="text" Text=" Driver={Microsoft Excel Driver (*.xls)};"/>
   <Segement Type="text" Text=" DriverId=790;"/>
   <Segement Type="text" Text=" Dbq="/>
   <Segement Type="File">
        <File Path="d:\Oblicore " NamePattern="Availabilty_2003_01.XLS">
   </Segement>
</ConnectionString>
```

Exemple 3 : Chaîne de connexion simple pour utilisation avec plusieurs fichiers Excel :

```
<ConnectionString>
   <Segement Type="text" Text=" pilote Excel de Driver={Microsoft (*.xls)};"/>
   <Segement Type="text" Text=" DriverId=790;"/>
   <Segement Type="text" Text=" Dbq="/>
   <Segement Type="File">
        <File Path="d:\Oblicore ",NamePattern="Availabilty_*.XLS"/>
   </Segement>
</ConnectionString>
```

Exemple 4 : Utilisation d'une entrée DSN d'ODBC standard :

une entrée de DSN d'ODBC standard permet de se connecter à toute source ayant une entrée de DSN créée dans le gestionnaire d'ODBC sur le serveur d'applications. Vous pouvez rechercher l'entrée de DSN d'ODBC standard dans la section des outils d'administration du serveur.

```
<ConnectionString>dsn=SampleDataSource;usr=scott;pwd=tiger;</ConnectionString>
```

Lecture des enregistrements d'une source de données de fichier

Tant qu'il existe une interface ODBC avec la source de données, il est possible de configurer un adaptateur SQL pour envoyer des requêtes sur des fichiers. Pour configurer l'adaptateur afin de lire plusieurs fichiers, il est nécessaire d'utiliser les éléments Segment dans l'élément ConnectionString. Vous trouverez un exemple dans la section précédente décrivant la chaîne de connexion.

Description de la façon dont l'adaptateur SQL fonctionne avec les fichiers :

1. Dans chaque requête, l'adaptateur tente de lire le fichier jusqu'à ce qu'il ne puisse plus récupérer d'enregistrements.
2. L'adaptateur passe alors au fichier suivant et essaye de le lire.
3. Lorsqu'il n'y a plus de fichiers, l'adaptateur se met en attente pour cette requête en fonction de l'attribut SleepTime.

Fichier Schema.ini

Lorsqu'un pilote de texte ODBC est employé, le format du fichier texte est déterminé par le fichier d'informations de schéma (schema.ini). Ce fichier d'informations de schéma doit être placé dans le même répertoire que la source de données de texte.

Les fichiers Schema.ini sont créés à partir d'entrées, chaque entrée décrivant la structure d'un fichier texte unique. La première ligne de chaque entrée est le nom du fichier source de texte, entouré par des crochets.

Lorsque l'adaptateur fonctionne avec des fichiers qui ont été définis à l'aide de caractères génériques, le nom de fichier change constamment. Comme le nom dans le fichier schema.ini ne peut pas contenir de caractères génériques, l'adaptateur doit modifier le fichier schema.ini avant de se connecter à la base de données.

Il est par conséquent nécessaire d'ajouter une ligne d'indication avant la ligne d'entrée du fichier. Cette ligne d'indication contient le modèle de nom de l'élément de chaîne de connexion dans le fichier de configuration de l'adaptateur, entouré par des crochets. L'adaptateur remplace la ligne suivante par le nouveau nom de fichier entouré par des crochets.

Remarque : Le fichier schema.ini peut contenir plusieurs entrées. Il est de votre responsabilité d'ajouter les lignes à l'emplacement correct.

Exemple de fichier Schema.ini

```
[sqltes*.txt]
```

```
[sqltest2.txt]
ColNameHeader = False
CharacterSet = ANSI
Format = TabDelimited
Col1=id Char Width 30
Col2=idname Char Width 30
----------------------------------------
[report_200*.txt]
[report_2003_09.txt]
ColNameHeader = False
CharacterSet = ANSI
Format = TabDelimited
Col1=date Char Width 30
Col2=service Char Width 30
Col3=responseTime Char Width 30
----------------------------------------
```

Fichier DataSourceControl

Pour chaque requête, le fichier de contrôle de la source de données contient le modèle de nom de fichier et le nom du fichier d'entrée actuel pour poursuivre avec le même fichier au redémarrage de l'adaptateur.

Voici un exemple de fichier de contrôle au format XML :

```
<AdapterControl Save="end" LastSaveTime="2005/03/23 09:16:15">
   <Data>
      <QueryCollection>
         <Query QueryName="TroubleTickets (on D:\Data\Incidents*.xls)">
            <KeyField Name="INC_CLOSE_DATE">
               <LastValue>03/07/05 13:06:21</LastValue>
            </KeyField>
            <LastFileName>IncidentsMarch.xls</LastFileName>
         </Query>
      </QueryCollection>
   </Data>
```

Conservation des fichiers d'entrée

Lorsque l'adaptateur a fini de lire le fichier actuel, il poursuit la recherche dans le suivant. Le prochain fichier à lire est le premier fichier qui correspond au masque de nom et dont le nom est supérieur (dans l'ordre lexicographique) au nom du fichier précédent. L'adaptateur ne rouvre pas les fichiers précédents, même si de nouveaux enregistrements ont été ajoutés à ces derniers.

L'adaptateur utilise l'attribut InitialFileName uniquement lorsque ces deux attributs sont définis sur non et que le fichier de contrôle ne contient pas le dernier nom de fichier.

Vérification des requêtes

L'adaptateur vérifie la chaîne de connexion et la requête uniquement si le fichier défini existe. S'il est défini avec des caractères génériques, l'adaptateur le vérifie uniquement par rapport au premier fichier.

Des erreurs peuvent se produire lorsque l'adaptateur tente de lire un nouveau fichier. Dans ce cas, l'adaptateur s'arrête immédiatement si l'attribut critical est défini sur yes. S'il est défini sur no, l'adaptateur ne continue pas l'exécution de cette requête, mais il poursuit par les autres requêtes. L'adaptateur peut aussi quitter le fichier actuel et passer au fichier suivant.

Variables internes de l'adaptateur

Deux variables internes peuvent être utilisées dans le fichier de configuration pour faire référence au contexte actuel du nom de fichier. Ce sont :file_name et :file_name_no_ext, qui renvoient au nom du fichier actuel, respectivement avec ou sans l'extension de fichier.

Vous pouvez utiliser ces variables dans les éléments SelectStatement et SelectInitialValues, ainsi que dans l'attribut QueryKeyField\Function.

L'adaptateur remplace la variable par le nom de fichier dans les requêtes.

Par exemple :

- select date, service, value from ":filename"
- select id and name from :file_name_no_ext

Créer un adaptateur à l'aide de l'interface utilisateur de CA Business Service Insight

Chaque adaptateur configuré pour s'exécuter dans l'environnement CA Business Service Insight doit être enregistré dans l'interface utilisateur, en plus d'être défini dans le registre. La principale raison est d'établir les paramètres pour l'écouteur d'adaptateur, afin qu'il soit prêt à recevoir les événements de l'adaptateur. Pendant cette étape, l'ensemble des paramètres de l'adaptateur, tels que les tables de conversion et les types d'événement, sont définis.

Procédez comme suit :

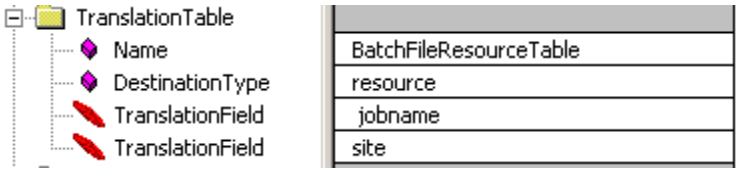
1. Créez l'adaptateur.
2. Choisissez des ressources/adaptateurs.
3. Vérifiez qu'aucun des adaptateurs existants dans la liste n'est défini sur le même port que votre adaptateur, c'est-à-dire sur le port qui est défini dans le fichier de configuration de l'adaptateur, dans le répertoire OblicoreInterface\OnlineInterface\Port.
4. Cliquez sur Ajouter, puis sélectionnez la méthode à utiliser pour créer l'adaptateur. Il existe diverses méthodes :
    a. **Créer manuellement** : cette option permet de configurer l'écouteur d'adaptateur pour se connecter à l'adaptateur que vous avez défini (ou allez définir) manuellement.
    b. **Utiliser l'assistant :** cette option permet de créer l'adaptateur à l'aide d'un assistant, étape par étape. Consultez la section suivante pour plus de détails.
    c. **A partir d'un modèle**
    d. **A partir d'un fichier de configuration existant** : cette option permet de charger un modèle d'adaptateur pré-configuré qui remplit automatiquement tous les champs de l'assistant de configuration d'adaptateur avec les options définies dans le fichier de configuration spécifié.
    e. La fenêtre résultante varie selon l'option choisie.
5. Remplissez les champs :

    Adresse réseau : saisissez l'adresse IP de l'adaptateur. S'il existe localement sur le serveur d'applications, indiquez localhost, sinon indiquez le port défini.
6. Cliquez sur Enregistrer.

Pour créer les tables de conversion nécessaires :

Remarque : Ces étapes doivent être accomplies pour chaque table de conversion définie dans le fichier de configuration :

1. Dans le menu Conception, cliquez sur Conversion, Tables de conversion et cliquez sur le bouton Ajouter.
2. Tous les paramètres de table de conversion doivent correspondre à la définition de table équivalente dans le fichier de configuration de l'adaptateur :
   - *Nom* : le nom doit correspondre à l'attribut Nom dans l'entrée Table de conversion, Nom du fichier de configuration.
   - Champs source : ce paramètre doit contenir tous les champs du champ TranslationField de la table de conversion. Ajoutez tous les champs. La valeur de conversion peut être le résultat d'une combinaison de deux ou plusieurs champs. Voici un exemple de résultat :



3. Type de destination : ici encore, ce champ doit correspondre à l'attribut DestinationType de la définition de table de conversion dans le fichier de configuration. (ressource, event_type, etc.).
4. Adaptateurs enregistrés : ajoutez les adaptateurs qui doivent utiliser cette table de conversion. Plusieurs adaptateurs peuvent utiliser une même table de conversion.
5. Cliquez sur Enregistrer.

6. Importez la définition des champs du type d'événement pour chaque type d'événement.

   Pour pouvoir importer la définition des champs d'un fichier de configuration d'adaptateur spécifique, cet adaptateur doit avoir été exécuté et avoir été connecté à CA Business Service Insight au moins une fois. Lorsque l'adaptateur se connecte à CA Business Service Insight, il transmet le fichier de configuration à CA Business Service Insight, ce qui permet au système d'utiliser sa définition de champs.

   Remarque : il est aussi possible de spécifier les définitions de champ manuellement en fonction du type d'événement.

7. Activez l'adaptateur :

   a. Dans le menu Conception, cliquez sur Collecte de données, Adaptateurs.

   b. Cliquez sur le bouton de démarrage de l'adaptateur.

Le tableau suivant contient les différents statuts des adaptateurs :

Statut	Description
Inactif	Etat initial. L'adaptateur est inactif et n'a pas encore été démarré.
Ecouteur inactif	Le service de l'écouteur d'adaptateur n'a pas démarré.
Démarrage en cours	L'adaptateur est en cours de démarrage.
Démarré	L'adaptateur a démarré.
Arrêt en cours	Arrêt en cours.
Pause en cours	Mise en pause.
En pause	Interrompu.
Pas en cours d'exécution	L'adaptateur ne s'exécute pas sur son ordinateur hôte.
Erreur	Erreur dans le fichier de configuration de l'adaptateur ; l'adaptateur ne peut pas démarrer.
Erreur de connexion	Erreur de connexion de l'adaptateur (erreur d'hôte ou de port), ou, avant l'exécution de l'adaptateur pour la première fois, aucun signal n'a été détecté. Statut lors de la première exécution de l'adaptateur.
Bloqué	Le nombre maximal de rejets d'événement a été atteint.

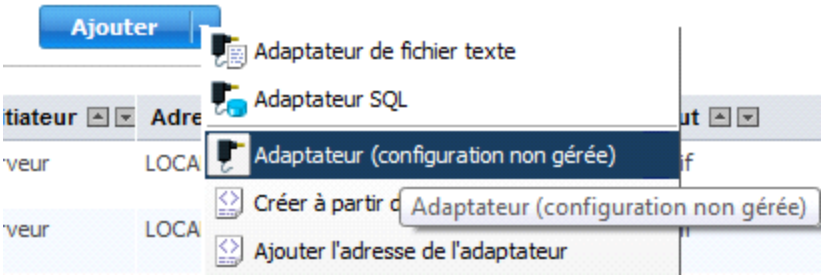
Créer un adaptateur à l'aide de l'assistant de configuration d'adaptateur

L'assistant de configuration d'adaptateur, une nouvelle fonctionnalité de l'interface utilisateur de CA Business Service Insight, facilite et guide la création d'un nouvel adaptateur avec une interface plus intuitive que l'éditeur XML. L'assistant vous guide dans une suite d'onglets contenant toutes les informations nécessaires à la création d'un adaptateur, et permet à la fin de télécharger une copie du fichier de configuration XML compilé. L'assistant comporte toutefois quelques restrictions.

Actuellement, il ne permet pas les actions suivantes :

- Faire référence à la même structure d'entrée (format d'entrée) à partir d'interfaces de source de données différentes
- Faire référence à la même structure de sortie (convertisseur) à partir d'entrées différentes
- Configurer un commutateur de format d'entrée et l'utiliser pour décider du format d'entrée à utiliser à partir de l'interface de source de données
- Fournir une définition d'un champ d'ID de la source de données dans la structure de sortie
- Fournir une définition de plusieurs fichiers dans la chaîne de connexion pour interroger des fichiers texte ou Excel avec la même requête
- Utiliser les contraintes d'heure UTC ou UtcNow
- Spécifier des valeurs commençant ou finissant par un espace

Lors de la création d'un nouvel adaptateur à l'aide de l'assistant, les quatre options présentées dans le tableau suivant sont disponibles :



Les deux premières options permettent de créer un adaptateur de fichier texte ou un adaptateur SQL à l'aide de l'interface de l'assistant. L'option suivante, Adaptateur (configuration non gérée) est l'option à choisir lors de l'ajout d'un adaptateur préconfiguré créé dans l'éditeur XML. Cette option ne permet pas de modifier la configuration de cet adaptateur ultérieurement à l'aide de l'assistant. La dernière option, Créer à partir du fichier de configuration, permet de transférer une configuration d'adaptateur pré-configurée et de la faire importer par le système dans l'interface de l'assistant pour finir de la modifier. Cette option nécessite qu'aucune des restrictions de l'assistant mentionnées ci-dessus ne soit implémentée dans cette configuration particulière.

En dehors de ce point, les options de configuration de l'assistant de configuration d'adaptateur offrent les mêmes fonctionnalités que celles décrites pour la configuration manuelle d'un adaptateur. Son but est de fournir une interface plus simple et plus accueillante pour modifier les paramètres de configuration. Les mêmes fonctions et principes que dans l'alternative manuelle s'appliquant ici, consultez les sections pertinentes pour tous détails.

Exécuter et tester l'adaptateur

En général, il n'est pas possible de terminer le paramétrage du fichier de configuration d'adaptateur en une seule fois. Il peut être nécessaire d'exécuter un certain nombre d'itérations d'exécution de l'adaptateur et de vérifier les résultats afin de s'assurer que la configuration de l'adaptateur est correcte.

Voici les principaux problèmes qu'il peut être nécessaire de corriger :

- Problèmes de connexion (entre la source de données et l'adaptateur, ou entre l'adaptateur et l'écouteur)
- Erreurs dans le fichier de configuration, par exemple :
  - Mauvaise structure
  - Mauvaise utilisation d'attributs
  - Mauvaise casse utilisée (l'adaptateur est sensible à la casse des caractères : Oui et oui sont différents, par exemple)
  - Mauvaise affectation de valeur
- Erreurs de manipulation de données, telles que structure d'événement de sortie, valeurs d'événement erronées, erreur dans les requêtes.

Procédez comme suit :

1. Paramétrez l'adaptateur avec RunOnce = "yes" et LogDebugMode = "yes", et donnez à l'attribut RejectedEventsUpperLimit une valeur raisonnable (voir Modes d'exécution de l'adaptateur (page 101)).

   L'illustration suivante montre les paramètres de configuration nécessaires pour les tests.
2. Il est également possible d'utiliser le mode hors ligne à des fins de paramétrage du fichier de configuration.
3. Une fois que le fichier de configuration s'est chargé correctement, redéfinissez ce paramètre sur en ligne (voir Modes d'exécution de l'adaptateur).
4. Chaque itération comprend les étapes suivantes :
   a. Mise à jour/correction du fichier de configuration de l'adaptateur.
   b. Supprimez tous les fichiers de sortie de l'adaptateur.
   c. Exécutez l'adaptateur en double-cliquant sur le raccourci vers son fichier exécutable *Adapter* ou sur le fichier *.bat* qui a été créé.

d. Ouvrez le fichier journal de l'adaptateur dans le navigateur de journaux (utilitaire Log Browser.exe, dans le répertoire Utilitaires de CA Business Service Insight) et vérifiez qu'il n'y a pas de messages d'erreur.

5. La première étape est d'obtenir un fichier de configuration correct et d'atteindre un état dans lequel l'adaptateur charge le fichier de configuration correctement. Répétez cette étape jusqu'à obtenir une connexion correcte à CA Business Service Insight et à la source de données et avoir rejeté les événements et demandes de conversion.
6. Pour terminer cette étape, vérifiez les points suivants :
   - Absence de messages d'erreur dans le fichier journal de l'adaptateur.
   - L'adaptateur se connecte correctement à CA Business Service Insight et à la source de données.
   - L'adaptateur a envoyé des requêtes de conversions et a rejeté des événements.

   La lettre "R" doit s'afficher sur la console pour tous les enregistrements rejetés par l'adaptateur. N'oubliez pas que le rejet d'événements est prévisible jusqu'à ce que toutes les conversions nécessaires soient terminées.
7. Vérifiez que le fichier rejectedEvents contient des enregistrements et n'est pas vide.

YGWorkflowAdapter
Adapter YGWorkflowAdapter
Running
<Connecting Oblicore>
LTT<Connecting DataSource (Commit) >
(Commit) RRRR

8. Connectez-vous à CA Business Service Insight, naviguez jusqu'à la page *Entrées de conversion* et recherchez des entrées de conversion en attente auprès de votre adaptateur. Vous devriez voir plusieurs entrées, une pour chaque demande de conversion envoyée par l'adaptateur.

   AVERTISSEMENT : la suppression des fichiers de sortie de l'adaptateur est une opération risquée. Elle doit être exécutée uniquement à cette étape, à des fins de tests. Par exemple, en cas de suppression du fichier de contrôle, l'adaptateur perd la liste des fichiers déjà lus et peut par conséquent perdre des données ou lire à nouveau des fichiers. Le seul fichier que vous pouvez supprimer pendant le mode opérationnel sans conséquences fonctionnelles est le fichier journal.

Pour utiliser le navigateur de journal afin d'afficher les messages d'erreur :

S'il existe un message d'erreur, double-cliquez sur ce message pour le lire. Ces messages sont habituellement causés par une erreur dans le fichier de configuration.

Conversions de ressources et d'événements

Lors de l'étape précédente, un certain nombre d'événements rejetés ont été créés pendant l'exécution de l'adaptateur. Ces événements rejetés ont été stockés dans un fichier RejectedEvents.txt , mais également comme entrées en attente de conversion dans la base de données de CA Business Service Insight. L'étape suivante du chargement des données brutes dans le système consiste à fournir une conversion de ce qui a fait l'objet d'une mesure pour permettre à CA Business Service Insight d'utiliser ces données selon les besoins.

Il est possible que des événements de données brutes soient rejetés à cause du type d'événement OU de l'absence de définition de la ressource dans le système. Les événements en attente sont définis par le type de table de conversion dont ils proviennent. Les exemples les plus courants sont des types d'événement provenant d'une table de conversion et des ressources provenant d'une autre table de conversion.

Lorsque l'adaptateur rencontre une nouvelle ressource (par exemple lorsqu'un nouveau serveur est ajouté dans l'outil de surveillance du réseau et apparaît comme nouvelle entrée dans la source de données de cet utilitaire), vous pouvez l'ajouter au modèle de ressource du système. Deux étapes sont nécessaires pour que cette nouvelle ressource soit acceptée dans CA Business Service Insight.

Vous devez d'abord créer la ressource comme entité CA Business Service Insight (une ressource), puis indiquer une conversion. Cette opération crée un lien entre la représentation de chaîne trouvée dans la source de données et l'entité définie comme ressource dans CA Business Service Insight. Cette opération en 2 étapes peut être accomplie en une seule fois via l'interface utilisateur, à l'aide d'un processus d'ajout et de conversion qui permet de créer la ressource et l'entrée de conversion nécessaire en une seule opération. Lors de l'ajout et de la conversion, plusieurs entrées peuvent être sélectionnées sous réserve qu'elles aient les mêmes paramètres d'allocation. La conversion est le processus de création de la correspondance entre la valeur de la source de données et l'entité CA Business Service Insight. Lors de la création des conversions, une entrée est ajoutée à la table de conversion avec les valeurs correspondantes. Lors des requêtes ultérieures envers la source de données, l'adaptateur saura alors automatiquement comment gérer cette nouvelle valeur.

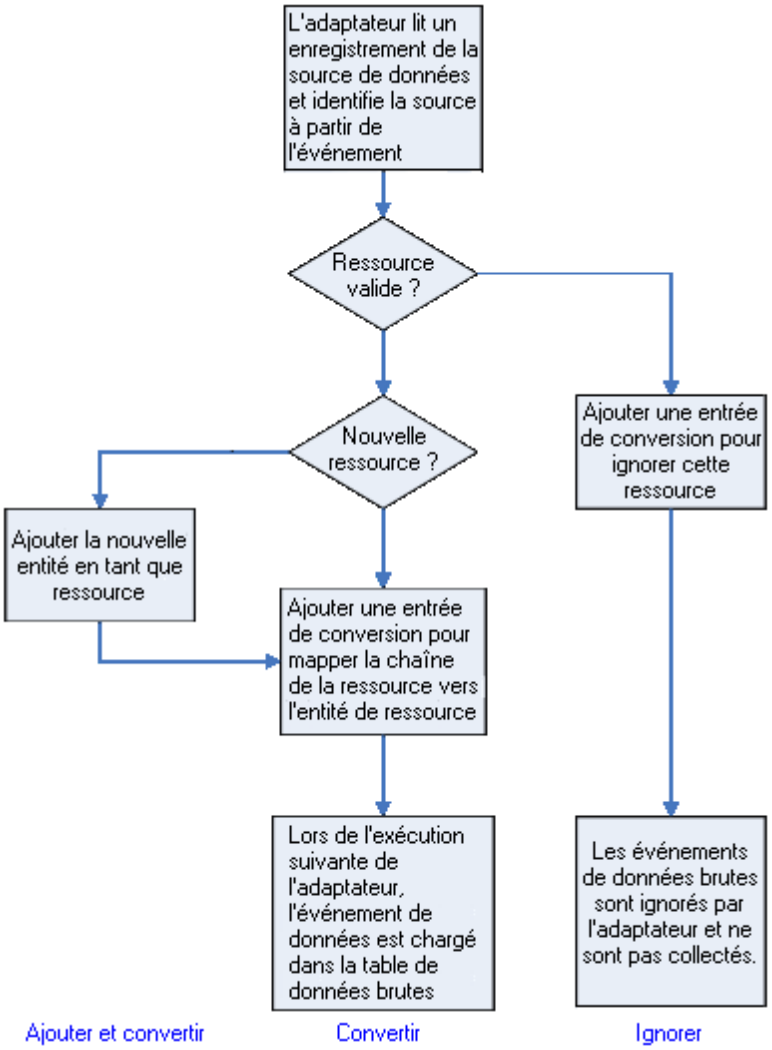
A ce stade, l'adaptateur a déjà été exécuté et a envoyé des demandes de conversion pour chaque valeur des champs Conversion. Les événements associés à ces valeurs ont été rejetés et attendent d'être envoyés à CA Business Service Insight lorsque la conversion sera terminée. Les conversions peuvent être effectuées manuellement ou automatiquement, à l'aide d'un script de conversion.

Les actions de conversion suivantes sont possibles pour les entrées de conversion en attente :

Convertir : Permet la création d'une entrée dans la table de conversion, en faisant la correspondance entre une valeur de source de données et l'entité de CA Business Service Insight appropriée. L'entité CA Business Service Insight vers laquelle les conversions sont faites doit déjà exister. (Le cas d'une ressource mal orthographiée d'une source de données en est un bon exemple. Des noms différents indiqués dans la source de données peuvent en fait référencer une même entité logique. Par exemple Server001 et SERVER001. Les ressources de CA Business Service Insight sont sensibles à la casse.)

- **Ajouter et convertir** : Permet la création d'une nouvelle entité dans CA Business Service Insight *et* ajoute simultanément une entrée de conversion pour cette entité dans la table de conversion. Il s'agit de l'action la plus courante accomplie sur des entrées de conversion en attente, car le mécanisme de conversion est utilisé pour créer l'infrastructure dans CA Business Service Insight.
- **Ignorer** : Lorsqu'une entrée est ignorée, tous les événements associés sont ignorés et ne sont pas envoyés à la table de données brutes de CA Business Service Insight. Les événements ignorés sont perdus. Par exemple, si la source de données contient des informations sur tous les serveurs d'un centre de données, mais que seules les données des serveurs d'applications sont nécessaires pour le calcul de niveau de service, tous les serveurs atteindront l'Adaptateur pour une conversion, mais seuls les serveurs d'applications seront convertis. Tous les autres sont ignorés puisque CA Business Service Insight fait en sorte que les événements inutiles soient aussi ignorés. Une entrée ignorée peut être convertie plus tard si nécessaire, mais ces données seront uniquement capturées à compter de ce moment.
- **Supprimer** : L'entrée de conversion est supprimée et l'événement rejeté associé est également supprimé. Si la même valeur est retransmise ultérieurement par la source de données, une nouvelle entrée en attente est créée.

L'organigramme suivant résume les cas d'utilisation de ces options :


L'adaptateur lit un enregistrement de la source de données et identifie la source à partir de l'événement
Ressource valide ?
Nouvelle ressource ?
Ajouter une entrée de conversion pour ignorer cette ressource
Ajouter la nouvelle entité en tant que ressource
Ajouter une entrée de conversion pour mapper la chaîne de la ressource vers l'entité de ressource
Lors de l'exécution suivante de l'adaptateur, l'événement de données est chargé dans la table de données brutes
Les événements de données brutes sont ignorés par l'adaptateur et ne sont pas collectés.
Ajouter et convertir
Convertir
Ignorer

Conversions manuelles

Les conversions manuelles sont nécessaires lorsque l'entité existe déjà dans CA Business Service Insight. Cela peut se produire dans diverses situations. Par exemple, lorsqu'un script de conversion destiné à créer automatiquement les ressources d'une source externe a été exécuté (mais qu'il a été impossible d'automatiser les conversions). Ou lorsqu'une source de données comporte des entrées douteuses et qu'une ressource a été définie deux fois de façons différentes (par exemple Server001p et server001P). Des ressources créées manuellement peuvent également en être la cause.

Pour exécuter une conversion manuelle lorsque la ressource existe déjà :

1. Dans le menu Conception, cliquez sur Conversions, Entrées de conversion.
2. Par défaut, toutes les entrées en attente s'affichent.
3. Sélectionnez l'entrée de conversion en attente à convertir, en activant la case à cocher en regard de cette entrée.
4. Cliquez sur Convertir.
5. Choisissez l'entité pertinente (Ressource, Type d'événement, etc.) dans la liste. (Si aucun élément n'apparaît dans la liste, il peut être nécessaire de changer les critères de recherche par défaut.)
6. Sélectionnez la ressource ou le type d'événement dans la liste des entités, en cliquant sur la ligne contenant l'élément. Il reste en surbrillance lorsqu'il est sélectionné.
7. Cliquez sur Convertir. L'entrée de conversion est maintenant stockée dans le système.

Pour exécuter une conversion manuelle lorsque la ressource n'existe pas encore :

1. Dans le menu Conception, cliquez sur Conversion, Entrées de conversion.
2. Par défaut, toutes les entrées en attente s'affichent.
3. Sélectionnez l'entrée de conversion en attente à transformer en ressource CA Business Service Insight et à convertir, en activant la case à cocher en regard de cette entrée.
4. Cliquez sur Ajouter & Convertir.
5. Assurez-vous que le nom de ressource est spécifié comme voulu. Vous pouvez également personnaliser le champ *Nom d'affichage* pour changer la façon dont la ressource s'affiche sur un rapport. Si cette ressource doit être gérée comme partie d'un ensemble de modifications, cet ensemble de modifications spécifique doit aussi être spécifié ici.
6. La date d'entrée en vigueur de la ressource doit être configurée comme la date à laquelle cette ressource doit pouvoir faire l'objet de rapports dans le système. Notez que la ressource n'apparaîtra pas dans des rapports antérieurs à la date spécifiée ici.
7. Cliquez sur l'onglet Détails et sélectionnez les options suivantes pour l'association de ressource : En vigueur (true/false), Type de ressource, Appartenance au groupe de ressources, Service et Association de contrats.
8. Cliquez sur Enregistrer et convertir. La ressource est ajoutée au catalogue de services de ressources et l'entrée de conversion est également stockée dans le système. Lorsque les entrées en attente ont toutes été traitées, vous pouvez vérifier que les données se chargent dans le système :
9. Naviguez jusqu'aux ressources dans CA Business Service Insight et vérifiez que la nouvelle ressource a été créée.
10. Exécutez à nouveau l'adaptateur.
11. Contrôlez que le fichier d'événements rejetés est vide, à partir de son contenu et de sa taille.
12. Contrôlez que le fichier d'événements transmis est vide, à partir de son contenu et de sa taille.
13. Vérifiez avec l'outil de données brutes que les événements ont été ajoutés aux données brutes.

Configuration de l'infrastructure

La configuration de l'infrastructure inclut la définition des entités de modélisation de données telles qu'elles ont été identifiées pendant la phase de conception.

Cette étape n'inclut pas la définition de toutes les entités d'infrastructure, et elle n'est complète que lorsque la configuration de l'adaptateur est terminée.

Pendant cette étape, les entités suivantes sont introduites dans le système CA Business Service Insight :

- Types d'événement
- Types de ressource
- Ressources et allocations de ressources
- Groupes de ressources

Remarque : une fois que l'adaptateur a été exécuté correctement, vous pouvez utiliser la fonctionnalité d'importation automatique lors de la définition des champs Type d'événement.

Conversions automatiques avec des scripts de conversion

La conversion automatique est l'automatisation des processus de création et de conversion et de l'infrastructure, à partir d'une source de données externe et à l'aide de scripts qui accomplissent les actions de conversion.

La conversion automatique est effectuée par des scripts de conversion. Les scripts de conversion accélèrent le processus de mappage des nouvelles ressources informatiques et métiers dans CA Business Service Insight. Le script de conversion identifie automatiquement une nouvelle entrée de conversion lorsqu'elle est reçue, et convertit les ressources afin de permettre un mappage des ressources rapide et efficace. L'automatisation prend en charge l'interface vers les CMDB, en permettant au système d'identifier des ressources d'après leur définition configurée. La conversion automatique offre de nombreux avantages, notamment la possibilité de faciliter la maintenance des conversions et de prévenir les erreurs. Vous pouvez utiliser des scripts de conversion pour créer de nouvelles ressources et allouer des modifications.

En outre, vous pouvez utiliser des scripts de conversion pour :

- Convertir des entrées en objets CA Business Service Insight existants.
- Ajouter de nouveaux objets CA Business Service Insight et convertir en ces derniers d'après des entrées de conversion existantes.
- Créer des objets et convertir des entrées sur la base de tables extérieures à CA Business Service Insight, par exemple des tables de ressources d'un autre CMS externe.
- Vérifier l'existence d'un objet.
- Créer des ressources et allouer des ressources, par exemple types de ressource, groupes de ressources, contractants, et composants de service.
- Allocation/désallocation de ressources à des ensembles de modifications.

Puisque le processus de conversion est également pris en charge comme procédure manuelle dans l'interface utilisateur, il est nécessaire de choisir un processus de conversion. Ce faisant, prenez en considération les avantages et inconvénients suivants de la conversion automatique :

- Les scripts de conversion entraînant une plus grande complexité des projets nécessitent du savoir-faire et des efforts de développement de scripts.
- Les délais de développement et d'assurance qualité (pour les tests) sont plus longs.
- L'implémentation n'est pas nécessaire lorsque le processus de conversion est un simple effort initial unique.
- L'implémentation peut être planifiée dans le cadre d'une approche de seconde phase.
- La maintenance continuelle est réduite.
- Les erreurs de conversion humaines sont réduites.
- Pour toutes autres informations sur la création et l'exécution de scripts de conversion, consultez le manuel du *SDK* dans le chapitre sur l'intégration de CMDB.

Rubriques avancées sur les fonctionnalités des adaptateurs

Les sujets suivants abordent des thèmes avancés de la fonctionnalité d'adaptateur.

Singularité d'événement

La singularité d'événement est un mécanisme d'adaptateur qui fournit un processus empêchant l'insertion de données brutes en double dans la table T_Raw_Data.

Sans la singularité d'événement, lorsque l'adaptateur s'exécute contre une source de données et charge des événements dans la base de données, aucune validation n'est accomplie pour un événement identique. La fonctionnalité de singularité d'événement résout ce problème en permettant de spécifier s'il faut ou non vérifier la singularité des événements avant leur insertion, et que faire si le cas se présente. Ce processus de vérification peut toutefois avoir un effet adverse sur les performances de l'adaptateur.

Cette solution permet à l'utilisateur de définir une clé basée sur les champs de l'événement. Cette clé représente le caractère unique de l'événement : si deux événements ont la même clé, ce sont des événements identiques.

Il est également possible de spécifier l'action à entreprendre si une clé dupliquée est détectée dans la base de données. Ces actions sont présentées ci-dessous.

Remarque : La clé peut être définie comme une combinaison de plusieurs champs du convertisseur.

Fichier de configuration d'adaptateur avec singularité d'événement

TranslatorCollection/Translator/OnDuplication

Ce champ détermine ce qui doit être fait si l'événement existe déjà dans la base de données.

Les valeurs possibles sont :

- **Add** : insère des événements (supplémentaires) dans la base de données.
- **Update** : provoque la suppression (dans certains cas) de l'événement existant et l'insertion d'un nouveau, après validation du changement d'événement.
- **updateAlways** : provoque la suppression (dans certains cas) de l'événement existant et l'insertion d'un nouveau sans vérifier l'existence de changements.
- **Ignore** : n'insère pas de nouvel événement dans la base de données.

La valeur par défaut est Add.

Comparaison entre Ignore et Add

Certaines baisses de performances mineures peuvent survenir pendant l'exécution de l'adaptateur en mode Ignore. Toutefois, en mode Add, le système déclenche un nouveau calcul dans chaque cas d'événement dupliqué.

Comparaison entre Update et updateAlways

L'option Update diminue les performances de l'adaptateur, alors que updateAlways diminue les performances de calcul en déclenchant un nouveau calcul à chaque cas.

TranslatorCollection > Translator > TranslatorFields > TranslatorField > IsKey

Cet attribut détermine si le champ auquel il appartient doit contribuer à sa valeur pour la clé unique pour les données brutes. Il est inclus si la valeur est yes et non inclus si sa valeur est no. La valeur par défaut (si aucune valeur n'est indiquée) est yes pour les champs standard (ResourceId, EventTypeId et Timestamp) et no pour tous les autres. La clé doit être choisie avec précaution pour que les données brutes préservent l'intégrité nécessaire et assurent la précision des calculs.

Comportement de l'écouteur d'adaptateur avec la singularité d'événement

Lors de la réception d'un nouvel événement de l'adaptateur, l'écouteur vérifie la valeur du champ OnDuplication. Lorsque la valeur est add, le processus d'insertion standard s'effectue. Lorsque la valeur n'est pas add, l'écouteur vérifie l'existence d'un événement avec la même UniqueKey et le même ID de lecteur dans la base de données. Si la base de données contient déjà un événement comme décrit, le nouvel événement n'est pas inséré dans la base de données lorsque la valeur de OnDuplication est ignore.

Lorsque la valeur de OnDuplication est update, un contrôle des changements dans l'événement est effectué. Si tous les champs sont identiques, le nouvel événement n'est pas inséré dans la base de données.

Lorsque la valeur de OnDuplication est updateAlways, le contrôle précédent est ignoré et une mise à jour se produit de toute façon.

En mode update et updateAlways, les étapes suivantes se succèdent :

- Si ResourceId, EventTypeId et Timestamp ont été modifiés, un nouveau calcul est fait pour toutes les métriques qui utilisent l'ancienne formule d'événement.
- L'événement est mis à jour avec les valeurs appropriées.

Rassembler les données de transaction

Les données de transaction sont souvent recueillies pour les comparer à des seuils ou pour calculer les pourcentages périodiques éventuels de réussite. Par exemple, toutes les cinq minutes une transaction virtuelle est effectuée avec un système et le résultat (temps de réponse en millisecondes), est enregistré comme illustré ci-dessous :

```
[…]
1/1/2004 00:00 432
1/1/2004 00:05 560
1/1/2004 00:10 329
1/1/2004 00:15 250
01/01/04 00:20 275
01/01/04 00:25 2860
01/01/04 00:30 140
[…]
```

Dans d'autres situations, au lieu d'utiliser des transactions virtuelles, il se peut que des accès aux transactions réelles aient lieu dans un système. Dans ces cas, des centaines voire des milliers de transactions peuvent être accomplies toutes les heures.

Dans les deux cas susmentionnés, le chargement d'un tel volume d'informations dans CA Business Service Insight doit être évité si possible.

Le rassemblement des données par périodes est la meilleure façon de réduire les quantités de données. Lorsque le seuil de réussite est fixé, vous pouvez accomplir l'agrégation en laissant l'adaptateur compter le nombre de transactions réussies dans la période ainsi agrégée. Par exemple, si le seuil de réussite dans l'exemple précédent est défini à 500 millisecondes, seules cinq transactions sur sept ont réussi dans la période affichée. Le problème avec cette approche est le seuil fixe : que se passera-t-il s'il est nécessaire de changer le seuil plus tard ? Dans une telle situation, les données brutes doivent être relues et testées par l'adaptateur avec le nouveau seuil.

Par conséquent, dans une situation optimale l'adaptateur doit agréger les données de transaction de manière flexible, sans perdre de données significatives.

Une solution limitée consiste à laisser l'adaptateur tester les transactions avec plusieurs seuils. Il existe deux méthodes pour ce faire :

- Un événement avec plusieurs tests : Event Type = {TotalTransactions, TransBelow250, TransBelow500, TransBelow750, […]}
- Plusieurs événements avec un test : Event Type = {TotalTransactions, Threshold, TransBelowThreshold}.

Les deux options présentent les mêmes problèmes : les seuils ne peuvent être changés à l'avenir que dans un petit ensemble de valeurs prédéfinies.

Solution recommandée

Hypothèse : les seuils potentiels sont des multiples d'un certain nombre. Par exemple, tous les seuils (en millisecondes) sont un multiple de 250. Ainsi, 0, 250, 500, 1750 et 3000 millisecondes sont donc tous des seuils potentiels.

Suivant cette hypothèse, la solution suggérée est d'agréger les transactions en arrondissant toutes les valeurs sur le multiple commun, et en comptant le nombre de transactions pour chaque valeur arrondie. Event Type = {RangeFrom, RangeTo, TransactionCount}

Par exemple, les événements suivants seront générés afin d'agréger les données affichées ci-dessus, où le multiple commun est 250 millisecondes :

```
{1,     250,    2}
{251,   500,    3}
{501,   750,    1}
{751,   1000,    1}
```

Commentaires :

L'horodatage de ces événements serait le même. Par exemple, tous les événements agrégés peuvent se produire le 01/01/07 à 00:00., et il peut y avoir un autre ensemble d'événements pour l'échantillon d'heure suivant, le 01/01/07 à 01:00, en supposant une agrégation horaire)

RangeTo est calculé en arrondissant la transaction au commun multiple (voir ci-dessous).

RangeFrom est égal à RangeTo moins le multiple, plus 1. Cette valeur est spécifiée pour des raisons de clarté uniquement, et vous pouvez l'ignorer.

Par exemple, une agrégation horaire s'afficherait comme suit (remplacez MULTIPLE par la valeur multiple) :

```
select trunc (time_stamp, 'hh') "TimeStamp",
        round (response_time/MULTIPLE, 0)*MULTIPLE-MULTIPLE+1 "RangeFrom",
        round (response_time/MULTIPLE, 0)*MULTIPLE "RangeTo",
        count (*) "TransactionCount"
from t_log
group by trunc (time_stamp, 'hh'),
        round (response_time/MULTIPLE, 0)*MULTIPLE
```

Dans la logique applicative, la condition suivante peut être appliquée aux événements :

```
If eventDetails("RangeTo")<=Threshold Then
```

```
        SuccessfulTransactions=SuccessfulTransactions+eventDetails("TransactionCo
unt")
End If
```

Quelques perspectives pour conclure :

- Puisque les transactions tendent à être distribuées normalement, le nombre d'événements agrégatifs doit être relativement faible.
- Le choix de multiples communs élevés provoquera moins d'événements agrégatifs.
- Le volume d'événements agrégatifs doit toujours être inférieur ou égal au volume de données brutes.

Exécuter un adaptateur derrière un pare-feu

Afin de pouvoir exécuter un adaptateur derrière un pare-feu, la solution suivante est suggérée :

Ouvrez deux ports. Le premier port est le port affecté à l'adaptateur dans CA Business Service Insight. Le deuxième port, qui est facultatif mais recommandé, est le port du serveur de journalisation. Le port du serveur de journalisation est défini par défaut comme port 4040. L'ouverture du port de serveur de journalisation permet à l'adaptateur de transmettre des rapports au journal de CA Business Service Insight et de générer des alertes. Cette option est facultative, car l'adaptateur transmet toujours des rapports à un fichier journal local.

Pour les deux ports, le protocole à utiliser est TCP/IP.

Charger des données à partir de plusieurs répertoires

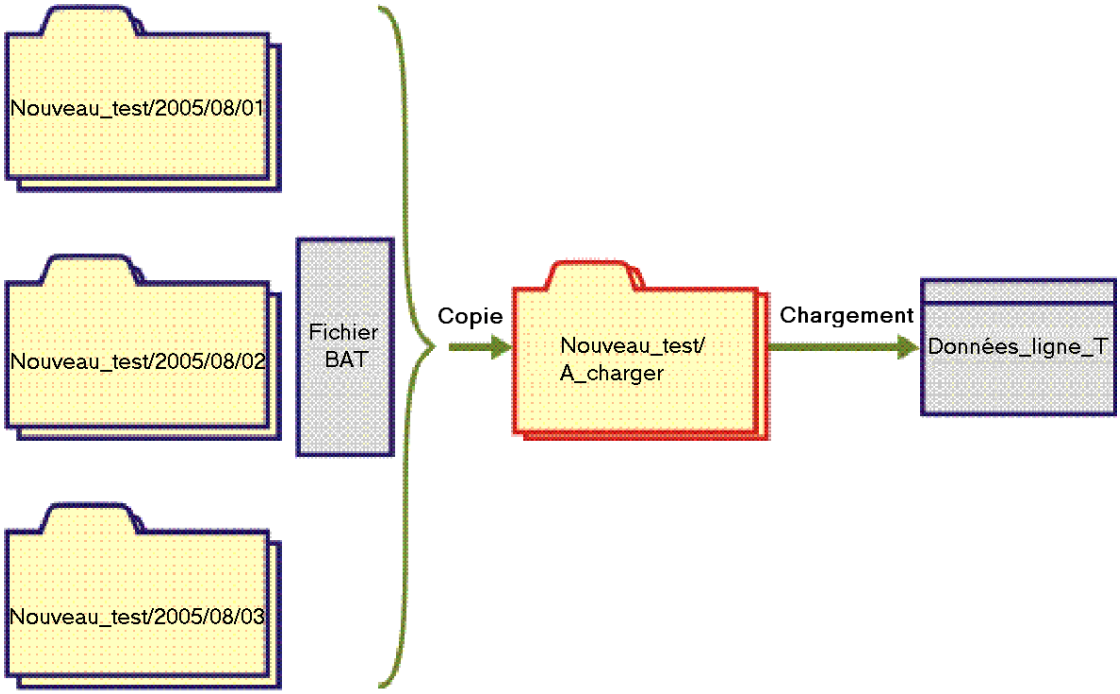
Cette section décrit une solution que vous pouvez mettre en oeuvre si les fichiers de source de données (en entrée d'un adaptateur de CA Business Service Insight) se trouvent chaque jour dans un répertoire différent, ou pour toutes les périodes définies (suivant une convention d'attribution de nom spécifique).

Par exemple, soit la structure de répertoires c:\NewData\AAAA\MM\JJ. Dans cet exemple, chaque jour correspond à un nouveau répertoire dans lequel les fichiers de ce jour sont localisés.

L'adaptateur de CA Business Service Insight doit parcourir le répertoire le plus récent et charger les nouveaux fichiers.

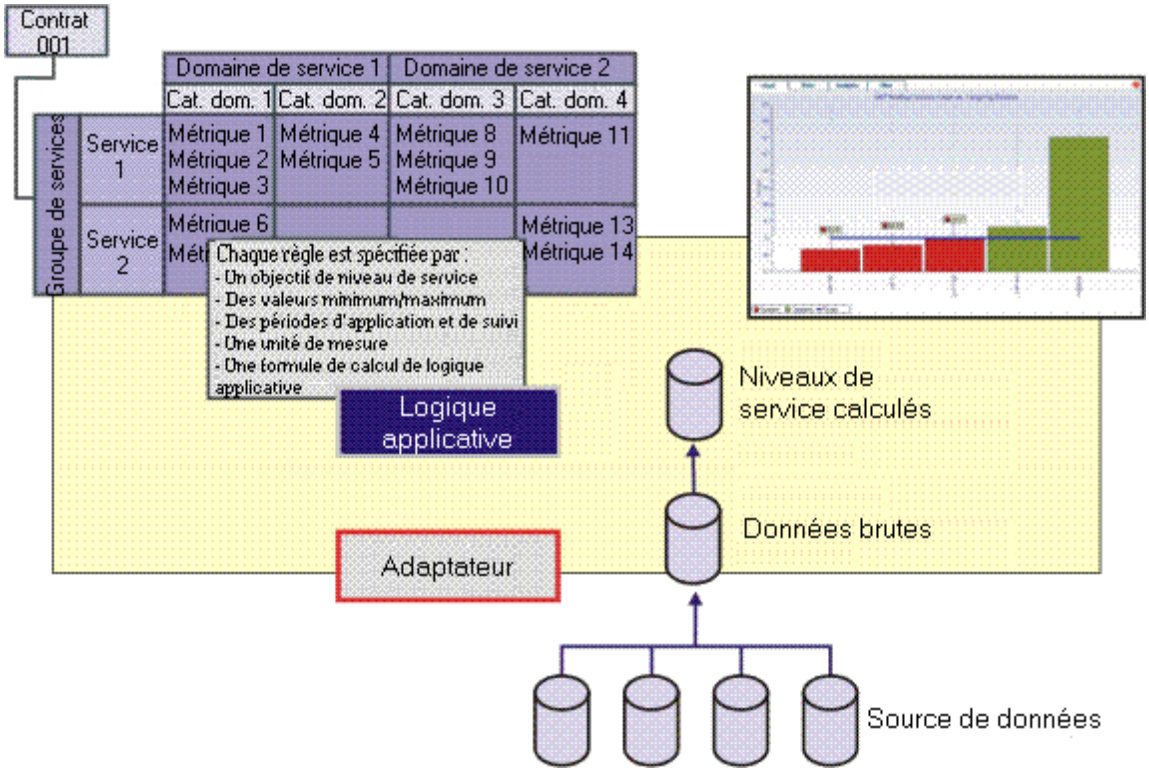
Une solution possible est d'ajouter quelques commandes au début du fichier .bat qui exécute l'adaptateur. Ces commandes copient les sources de données à charger depuis le dossier spécifique (d'après les conventions) dans un dossier dédié unique, créé spécialement à cette fin. L'adaptateur charge toujours les informations depuis ce dossier.

L'image suivante décrit cette solution :



Ecriture de scripts de logique applicative (expert en logique applicative)

L'image suivante décrit la position de la logique applicative dans CA Business Service Insight. Elle apparaît sous chaque métrique dans les contrats.



A ce stade de la mise en oeuvre, les adaptateurs pertinents sont configurés et les enregistrements de données brutes doivent déjà être disponibles dans la table T_RAW_DATA dans CA Business Service Insight. Il est maintenant possible d'appliquer la logique applicative aux événements afin de produire le résultat de niveau de service réel pour chaque métrique.

L'écriture de scripts de logique applicative est le processus d'écriture de code qui exécute des opérations logiques sur les données brutes (envoyées par les adaptateurs) afin de calculer les niveaux de service.

Chaque métrique possède sa propre logique applicative qui permet de calculer le niveau de service réel, même si la plupart des métriques du système partagent généralement une logique commune qui peut être appliquée à différents ensembles d'événements issus de données brutes.

Par exemple, une métrique qui calcule le délai de résolution des tickets de sévérité 1 et une autre métrique qui calcule le délai de résolution des tickets de sévérité 2 évaluent des ensembles différents d'enregistrements : l'une utilise uniquement les tickets de sévérité 1, et l'autre uniquement les tickets de sévérité 2. Toutefois, les deux appliqueront probablement la même méthode pour calculer le temps de résolution. Le script de temps de résolution sera développé et testé une fois, défini comme module de logique applicative, et sera ensuite utilisé par les deux métriques en incluant ce module dans les zones de logique applicative des métriques.

Par conséquent, lors du développement de scripts de logique applicative, les modules principaux ou les modèles sont habituellement développés de façon à être mis à la disposition de toutes les métriques du système. De plus, chaque catégorie de domaine reflète habituellement un type différent de mesure et par conséquent chaque catégorie de domaine suit généralement un module ou un modèle de logique applicative différent.

Flux de travaux de l'écriture de scripts de logique applicative

L'étape d'écriture de scripts de logique applicative implique les étapes suivantes :

- Définition d'une formule

  Créez la formule d'après les besoins de calcul définis lors de la phase de conception. Les formules définies sont toutes les formules uniques à utiliser dans leurs diverses permutations dans les métriques des contrats, chacune sous forme de module de logique applicative.

  Par exemple, si le contrat contient trois métriques pour calculer le temps de résolution moyen des tickets et une métrique pour chaque priorité de ticket, une formule unique est développée pour calculer le temps de résolution des tickets, et elle reçoit comme paramètre la priorité des tickets. Une fois testée, cette formule est définie comme module et liée à toutes les métriques appropriées.

- Test de la formule

  Les tests permettent ensuite de s'assurer que la formule est définie correctement et exempte d'erreurs, et que les calculs produisent le résultat attendu. Il est important de couvrir tous les cas extrêmes et les situations de limite dans le cadre des tests. La formule est exécutée pour les besoins du test dans le cadre de la logique applicative. Une fois qu'elle a été définie, la formule est testée dans son intégralité. Lorsqu'elle est appliquée à toutes les métriques comme module, il est important d'exécuter chaque métrique dans ce cadre au moins une fois pour vérifier qu'elle reçoit les événements (c'est-à-dire que l'enregistrement est correct) et produit un résultat raisonnable.

- Définition du module de SLALOM associé

  Chaque module représente un calcul de logique applicative unique. De plus, chaque module et sa définition de paramètre peuvent être associés à toutes les métriques appropriées. Pendant la définition du module, il est important que le module soit consciencieusement testé et documenté en détail : ce que fait le module (description du calcul), les paramètres qu'il attend (nom, signification et utilisation), etc.

- Association de toutes les métriques au module de logique applicative approprié

Vous devez définir un lien vers le module de logique applicative approprié pour chaque métrique appartenant aux contrats définis. Le module doit ensuite être exécuté dans la portée de la logique applicative pour s'assurer que le lien a été mis en oeuvre correctement et que l'enregistrement fonctionne correctement, en recevant les événements pertinents et en produisant les résultats attendus.

Modules de logique applicative

Il est important de prendre certains points en compte lors de la création de modules pour la logique applicative, particulièrement si vous utilisez plusieurs modules dans une métrique unique :

- Pour vous assurer que l'utilisation d'un module est claire, ajoutez un commentaire au début de la logique applicative pour cette métrique.
- Si vous utilisez quelques lignes de code dans l'espace de logique applicative de la métrique, et si vous incluez des modules dans ce code, veillez à supprimer cette section de code de la logique applicative de la métrique principale pour tout gestionnaire d'événements ou toute sous-routine par défaut. Assurez-vous que chaque sous-routine et chaque gestionnaire d'événements ne sont définis qu'*une fois* dans chacun des modules utilisés par une métrique particulière. Cette précaution évite toute confusion pour le moteur de calcul et la production de résultats inattendus.

  Remarque : Si, par exemple, vous définissez la fonction OnPeriodStart() dans votre module et y placez du code spécifique, puis laissez la fonction OnPeriodStart() par défaut sans code dans la fenêtre de la logique applicative principale de votre métrique, lors de l'exécution, le moteur ne saura pas quelle sous-routine exécuter. Il se peut qu'il n'exécute pas le code prévu.
- Procédez avec beaucoup d'attention si vous paramétrez la sous-routine OnRegistration. Dans certains cas, cette action peut interrompre le déclencheur automatique créé dans le système pour recalculer les métriques d'après l'ajout de nouvelles données brutes.

Détails sur la logique applicative

Toute la logique applicative est localisée dans un script en VBScript déclenché par des événements. Chaque événement qui atteint le logique applicative déclenche un gestionnaire d'événements.

Les types d'événement suivants sont envoyés par le moteur pour être évalués par la logique applicative :

- Evénements de données brutes. Entrées de données brutes réelles sur lesquelles la logique applicative base ses résultats. Le moteur envoie uniquement les événements de données brutes pertinents, d'après l'enregistrement de la formule.
- Evénements de moteur. Evénements envoyés implicitement par le moteur qui reflètent les propriétés de la définition de métrique, par exemple définitions de la période d'application et de la période de suivi.
- Evénements intermédiaires. Les événements générés par d'autres scripts de logique applicative peuvent être utilisés au sein d'un autre.

La formule de logique applicative contenue dans la définition de la métrique est ce qui évalue les événements et produit un résultat de niveau de service sur lequel les rapports sont basés. Selon ces résultats de niveau de service et la définition de domaine, le moteur produit également un résultat de déviation (si une cible de niveau de service a été appliquée à la métrique). Les résultats produits sont stockés dans une table de base de données appelée T_PSL. Cette table est interrogée par l'assistant de création de rapports lorsqu'il crée les rapports. Toutes les données des rapports sont donc précalculées afin d'optimiser les performances de génération de rapports.

Flux d'événements

Comme nous l'avons mentionné, les entrées de la logique applicative sont les événements de moteur et les événements de données brutes.

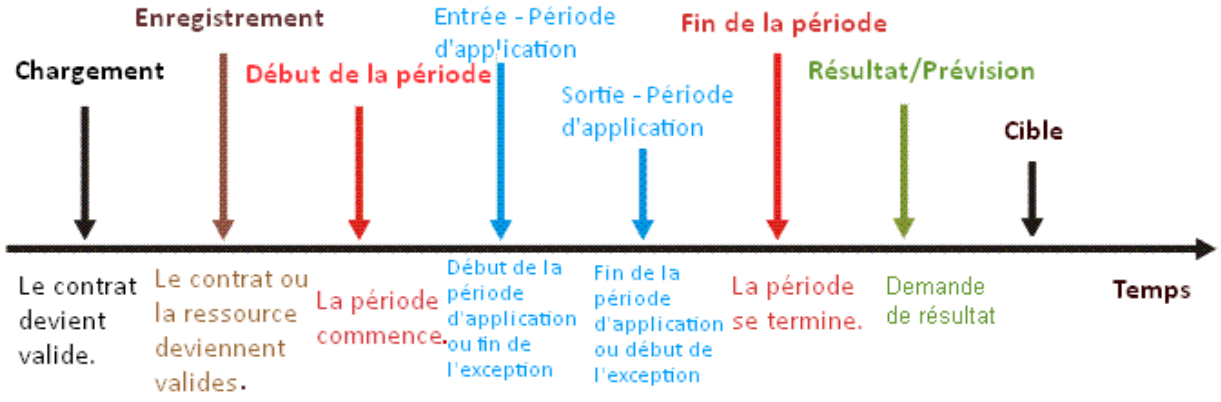
Les événements de données brutes reçus par la logique applicative sont déterminés par la fonction d'enregistrement, dans laquelle le code demande une définition spécifique des événements de données brutes définis par leur type d'événement et leur identificateur de ressource.

Dans la logique applicative, l'enregistrement associe également une sous-routine définie par l'utilisateur, qui est exécutée afin de gérer l'événement de données brutes une fois qu'un événement est reçu. (Valeur par défaut, c'est l'événement OnXXXEvent, qui doit être renommé de façon plus explicite.)

Les événements de moteur sont déclenchés par le moteur d'après les définitions des contrats et les métriques associées. Lorsque l'événement de moteur est déclenché et reçu, le moteur exécute le gestionnaire d'événements pertinent. Chaque événement de moteur possède un gestionnaire d'événements implicite. Ces gestionnaires d'événements sont des fonctions et des procédures définies au-dessus de VBScript. Le gestionnaire d'événements qui maîtrise l'enregistrement et la fonction result doivent obligatoirement être implémentés dans le code. Tous les autres gestionnaires d'événements sont facultatifs. Toutefois, la logique applicative ne s'occupe pas des événements de moteur pour lesquels les gestionnaires d'événements ne sont pas implémentés. Par conséquent, il est recommandé de les laisser tous en place (même s'ils ne sont pas utilisés) en prévision d'améliorations futures.

Remarque : Lors de l'écriture du script de logique applicative, il est important d'implémenter tous les événements du moteur afin de couvrir toutes les possibilités finales. Par exemple, même si pendant la première étape de l'implémentation une définition de période d'application n'était pas applicable, mais le sera dans l'avenir, toutes les formules nécessiteront une modification. Il est par conséquent recommandé que l'expert en logique applicative définisse d'avance le comportement durant la période d'application et en dehors de la période d'application, même si initialement ces définitions ne sont pas applicables, afin de réduire les modifications nécessaires lorsque le comportement sera introduit.

Voici divers événements de moteur et leurs gestionnaires d'événements :



- **OnLoad (Time)** (facultatif) : appelé une fois au démarrage des calculs lorsque le contrat est activé. Vous pouvez l'utiliser pour initialiser des variables globales.
- **OnRegistration (Dispatcher)** (obligatoire) : procédure permettant de demander les événements de données brutes pertinents et de les associer aux procédures définies par l'utilisateur avec lesquelles elles doivent être gérées. Les événements sont interrogés et associés à des procédures en utilisant les méthodes de l'objet Dispatcher.
  OnRegistration est appelée une première fois par le moteur de calcul au début du calcul de la métrique, et à nouveau chaque fois qu'une ressource enregistrée pour la métrique prend effet, après quoi le moteur évalue l'ensemble des changements effectués pour cette ressource qui peuvent influencer l'ensemble d'événements transmis à la formule. Le moteur fait définir la demande d'événement par le type d'événement et les identificateurs de ressources, et dans le cas où une ressource (ou un ensemble de modifications contenant plusieurs ressources) modifie des éléments relatifs à cet ensemble. Une fois qu'il entre en vigueur, un gestionnaire d'événements OnRegistration est déclenché.
- **OnPeriodStart (Time)** (facultatif) : appelé au début de la période d'agent (défini selon l'unité de temps). Le premier OnPeriodStart est déclenché une fois que le contrat entre en vigueur, lorsque l'équilibre des périodes commence par des unités de temps Ce gestionnaire d'événements est habituellement utilisé pour initialiser périodiquement des variables comme un compteur qui contient le nombre d'incidents ouvert dans la période de calcul qui devrait ensuite être initialisée à zéro au début de chaque période.

- **OnPeriodEnd (Time,IsCompleteRecord)** (facultatif) : appelé au début de la période d'agent (défini selon l'unité de temps). Il est toujours appelé à la fin arrondie de l'unité d'heure dans des heures arrondies (par exemple, 24:00). IsCompleteRecord est vrai lorsque la période de la métrique est terminée (en fonction de l'heure réelle du serveur d'applications), et est faux dans le cadre d'un calcul intermédiaire. Ce gestionnaire d'événements est habituellement utilisé pour effectuer les derniers calculs pour la période de fin afin de préparer au résultat final qui sera fourni par la fonction Result.
- **OnTimeslotEnter (Time)** (facultatif) : appelé lors de l'entrée dans une période d'application basée sur la définition de métrique associée. Par exemple, si la métrique est associée à une définition de période d'application des heures ouvrables où chaque jour à 08:00 heures du matin correspond au début de période d'application, dans ce cas ce gestionnaire d'événements sera déclenché tous les jours à cette heure.
- **OnTimeSlotExit (Time)** (facultatif) : appelé lors de l'entrée dans une période d'application basée sur la définition de métrique associée. Par exemple, si la métrique est associée à une définition de période d'application des heures ouvrables où chaque jour se termine à 17:00 heures, dans ce cas ce gestionnaire d'événements sera déclenché tous les jours à cette heure.
- **Target()** (facultatif) : appelé lorsque la métrique est spécifiée comme ayant une cible dynamique. Cela vous permet de déterminer la cible de niveau de service d'une métrique pendant l'exécution de la logique applicative. Ces cibles peuvent inclure la consommation de composants de service ou des changements d'infrastructure. Il contient quatre valeurs, comme la fonction Result, soit une pour chaque mode. Cette fonction est exécutée *après* la fonction Result au cours de l'exécution normale.
- **Forecast()** (facultatif) : appelé une fois au cours d'une action de validation de la version de contrat en faisant calculer une fois par le moteur de calcul le contrat en mode de prévision. Le mode de prévision exécute un cycle de calcul complet sur le contrat (de la date de début à la date de fin de la version du contrat). Ce cycle n'a pour seul objet que de calculer les valeurs prévisionnelles. (Aucun calcul n'est effectué sur la table T_RAW_DATA table). Cette fonction est appelée *à la place* de la fonction Result au cours de ce mode d'exécution.
- **Result()** (obligatoire) : appelé par le moteur de calcul pour obtenir le résultat du calcul pendant une période donnée. La fonction Result est toujours appelée après le gestionnaire d'événements OnPeriodEnd.

Les éléments suivants sont les étapes suivies par le mécanisme de calcul (service PslWriter) pour traiter une formule de calcul de logique applicative unique :

- PslWriter charge la formule dans la mémoire et exécute toutes les instructions dans la section de déclarations (la section de déclarations comprend tout le code situé à l'extérieur d'une fonction ou d'une sous-routine). Notez également que tous les modules et les bibliothèques joints sont inclus et compilés dans ce module de code unique pour l'exécution.
- PslWriter appelle le gestionnaire d'événements OnLoad.
- PslWriter appelle OnRegistration.
- Le calcul de la période commence par l'appel à OnPeriodStart.
- Pour chaque événement de données brutes enregistré pendant OnRegistration qui correspond à la plage horaire de la période, PslWriter appelle le gestionnaire d'événements défini par l'utilisateur associé à cet événement.
- Si le début ou la fin de la période d'application correspond à la plage horaire de la période, OnTimeslotEnter ou OnTimeSlotExit sera appelé.
- S'il y a un changement d'infrastructure pertinent dans la plage horaire de la période, OnRegistration sera appelé.
- Le calcul de la période commence par l'appel d'OnPeriodEnd.
- Si une cible dynamique a été spécifiée, cette fonction sera appelée.
- PslWriter appelle la fonction Result afin de recevoir le résultat final du calcul de période.

  Remarque : Lorsque la version de contrat est d'abord validée et qu'une prévision a été sélectionnée, la fonction Forecast est appelée au lieu de la fonction de résultat. Cette opération, toutefois, se produit uniquement dans ce cas pour chaque version)

Pendant chaque cycle de calcul, le moteur évalue quels sont les événements de moteur et les événements de données brutes pertinents en fonction de la période de calcul. Il va les trier d'abord par heure, puis les envoyer aux formules pertinentes pour être calculés. L'heure de l'événement de données brutes est son horodatage et le moment de l'événement de moteur est son heure de déclenchement. Ces deux types d'événement sont alors combinés dans une séquence de temps et envoyés pour être calculés.

Les périodes des événements sont basées sur la métrique locale associée, mais le paramètre Time des gestionnaires d'événements (c.-à-d., OnPeriodStart (Time)) et l'horodatage de l'événement de données brutes sont basés sur leur valeur UTC. Le moteur les compare en fonction de leurs valeurs UTC pour utiliser un point fixe de référence.

Exemple **:**

Si la différence UTC de fuseau horaire d'une métrique est de deux heures (c.-à-d. GMT+2), et que la valeur d'horodatage d'un événement d'ouverture d'incident est 10:00 heures du matin, l'horodatage utilisé par le gestionnaire d'événements dans le moteur sera en fait déplacé en conséquence et démarrera réellement à 8:00 heures du matin UTC. En supposant que l'adaptateur est configuré pour utiliser ce même fuseau horaire, les événements de données brutes seront également reculés de 2 heures UTC dans la base de données. Lorsque les événements sont transmis à la logique applicative, l'agent de calcul chargé du calcul des événements pendant la période démarrant à 10:00 heures du matin utilisera en fait l'heure UTC pour les événements, c-à-d 8:00 heures du matin. Toutefois, si vous utilisez le message out.log dans le code pour imprimer les horodatages, il indiquera que l'horodatage a adopté le temps UTC et affichera donc 08:00 heures du matin, bien que la période spécifiée (selon la métrique) est 10 heures du matin.

Dans la condition de calcul suivante, il est important de convertir l'horodatage de l'événement avant de l'utiliser :

Si une métrique est de calculer le nombre d'incidents qui ont été clôturés le même jour de leur ouverture, il est nécessaire de comparer l'heure d'ouverture et l'heure de fermeture de chaque incident. Si l'ouverture d'un incident et l'heure de fermeture correspondent au même jour (et dans une période d'application définie), l'incident sera pris en compte.

Il est possible que durant le processus d'adoption de l'heure UTC par l'incident à partir de son heure locale d'origine, le jour change. (c.-à-d. utilisation de GMT+2). Un incident ouvert à 1 heure du matin sera reculé à 23 heures UTC du jour précédent). Il ne sera alors pas compté même s'il devait l'être. Il s'agit d'une situation où le temps doit être reconverti en heure métrique locale, puis seulement alors comparé. Dans ce cas, utilisez la méthode GetLocaleTime(utcTime). Cette méthode convertit une heure donnée dans un fuseau horaire UTC en un fuseau horaire de la métrique locale.

```
Les informations suivantes sont le code de gestionnaire d'événements :
Sub OnIncidentEvent(eventDetails)
   If dateDiff("d",Tools.GetLocaleTime(eventDetails.time),_
     Tools.GetLocaleTime(eventDetails("ClosedAt")))=0 then
          CountIncidents=CountIncidents+1
   End If
End sub
```

Enregistrement

L'enregistrement est le processus au cours duquel la logique applicative soumet une demande au moteur de calcul pour l'ensemble des événements de données brutes qui doivent intégrer le calcul.

Vous pouvez maîtriser le processus d'enregistrement de deux façons : avec l'assistant d'enregistrement ou en utilisant manuellement l'objet Dispatcher dans la logique applicative.

L'utilisation de l'assistant d'enregistrement consiste simplement à choisir entre les options disponibles. Vous disposez de toutes les mêmes options que lorsque vous effectuez l'enregistrement manuellement, sans la possibilité d'utiliser des paramètres. Si vous avez besoin d'utiliser des paramètres, vous devrez effectuer l'enregistrement manuellement. Le flux de base de l'assistant nécessite de déterminer d'abord quel type d'enregistrement vous souhaitez accomplir, puis de définir les types de ressource et d'événements dans lesquels l'enregistrement doit être accompli, et finalement quel gestionnaire d'événements sera utilisé pour traiter les événements rassemblés.

Une fois que les enregistrements sont terminés, ils sont répertoriés dans l'onglet Enregistrement de la métrique. Notez également qu'il est possible d'avoir plusieurs instructions d'enregistrement pour une métrique.

En fait, l'assistant d'enregistrement utilise la même fonctionnalité que l'enregistrement manuel, et toutes ces options sont traitées dans la section suivante.

Dans le cas d'un enregistrement manuel dans la logique applicative, l'enregistrement de la formule est traité par le gestionnaire d'événements *OnRegistration.* Cette opération doit être implémentée dans la formule et déclenchée quand un événement de moteur d'enregistrement est déclenché. L'événement d'enregistrement est déclenché une fois, lorsque le contrat est activé, puis à chaque fois qu'une ressource pertinente ou l'ensemble de modifications est activé. Un changement dans la ressource concernée est considéré comme pertinent s'il affecte les événements que la métrique est supposée recevoir. Par exemple, si l'enregistrement est effectué par un contractant (*RegisterByContractParty*), cela signifie que tous les événements du type défini dont les ressources sont attachées au contractant de la métrique font partie du calcul. Dans ce cas, à chaque fois qu'une nouvelle ressource est attachée ou détachée à ce contractant, la méthode d'enregistrement sera déclenchée pour notifier le moteur du changement.

Les méthodes d'enregistrement sont fournies par l'objet Dispatcher qui est transféré à OnRegistration comme argument. Des méthodes différentes fournissent plusieurs méthodes pour définir les critères de filtre selon la définition de type d'événement et un critère d'allocation de ressources, comme les ressources d'un groupe de ressources, ou les ressources d'un certain type.

L'utilisation du contractant et des méthodes d'enregistrement de service est fortement recommandée parce qu'elle facilite l'utilisation de la logique applicative en tant que module ou modèle. De cette manière, le contractant et le service pertinents sont extraits de la définition de la métrique associée et lors de la ré-utilisation de la formule pour des contrats et/ou des composants de service différents, il n'est pas nécessaire de modifier l'enregistrement.

Une autre méthode d'enregistrement courante est RegisterByResourceGroup qui est utile pour travailler avec des ressources qui sont logiquement regroupées mais qui ne sont pas toujours associées à des contractants ou à des services. L'affectation de ressources aux groupes dans ce contexte peut être prise en charge par le catalogue de ressources (individuellement ou via des ensembles de modifications) et pourrait même être mise à jour automatiquement par des scripts de conversion.

En général, la méthode d'enregistrement est déterminée pendant la phase de conception et est directement pilotée par le modèle de données défini.

Remarque : Afin de vérifier si l'objet Dispatcher a correctement été utilisé, la fonction OnRegistration est également appelée pendant la vérification de syntaxe du SLALOM. Pour cette raison, il ne faut pas supposer qu'OnLoad a été exécuté avant la fonction OnRegistration et il ne faut pas utiliser certaines propriétés de l'objet de contexte, telles que TimeUnit, IntervalLength, IsPeriod, CorrectionsApply, et ExceptionsApply dans l'agent d'événement OnRegistration.

Les méthodes d'enregistrement sont également chargées d'associer les événements à une procédure qui sera déclenchée en fonction de l'horodatage de l'événement. La procédure définie peut avoir un nom, mais aura toujours l'objet eventDetails pour paramètre. L'objet eventDetails reflète l'événement de données brutes reçu et contient tous les détails d'événement comme des champs de données, comme indiqué dans l'enregistrement suivant :

```
Sub OnRegistration(dispatcher)
dispatcher.RegisterByContractPartyAndService "OnMemUseEvent", "MemUse", "Server"
'the parameters of the method are: <procedure name>, <Event Type>, <Resource
Type>
End Sub
```

L'instruction d'enregistrement ci-dessus nous indique que tous les événements de données brutes du type d'événement MemUse, et associé au type de ressource Server, sera envoyée au gestionnaire d'événements OnMemUseEvent dans la logique applicative.

La procédure suivante devra également être définie en amont dans la logique applicative :

```
Sub OnMemUseEvent(eventDetails)
 If InTimeSlot And eventDetails("MemoryUsage")>MaxMemoryUse Then
       MaxMsmoryUse = eventDetails("MemoryUsage)"
     End If
End Sub
```

En faisant référence à l'objet eventDetails et en utilisant le paramètre MemoryUsage, l'instruction extrait la valeur du champ MemoryUsage de l'événement qui a été passé dans la fonction. Ces champs sont les mêmes que définis dans le type d'événement, et les noms de champ respectent la casse.

Métriques groupées de registre

Les métriques groupées permettent à la définition d'une métrique de s'exécuter pour chaque membre d'un groupe de ressources, pour appliquer la même définition et logique à un ensemble d'éléments. Un groupement peut être défini statiquement sur un ensemble prédéfini de ressources ou dynamiquement sur les membres du groupe de ressources, tandis que le groupe peut être modifié dans le temps et inclure ou exclure des membres.

Remarque : Pour une description détaillée, consultez l'annexe E : Définition des formules de logique applicative (expert en logique applicative).

Les métriques groupées sont utilisées lorsqu'il est nécessaire de calculer un résultat de niveau de service pour chaque élément dans un groupe de ressources. Les éléments dans un groupe de ressources peuvent être des ressources ou d'autres groupes de ressources, par conséquent la méthode d'enregistrement dans un script de logique applicative d'une métrique groupée doit être RegisterByResourceGroup ou RegisterByResource, dans laquelle le nom de ressource ou le nom de groupe de ressources spécifié est défini comme l'élément dans le groupe. Cette opération s'effectue à l'aide de la propriété ClusterItem de l'objet de contexte qui contient le nom de l'élément de groupe actuel.

Exemple **:**

```
dispatcher.RegisterByResource  “<ProcedureName>”, “<Event Type name>”,
Context.ClusterItem
```

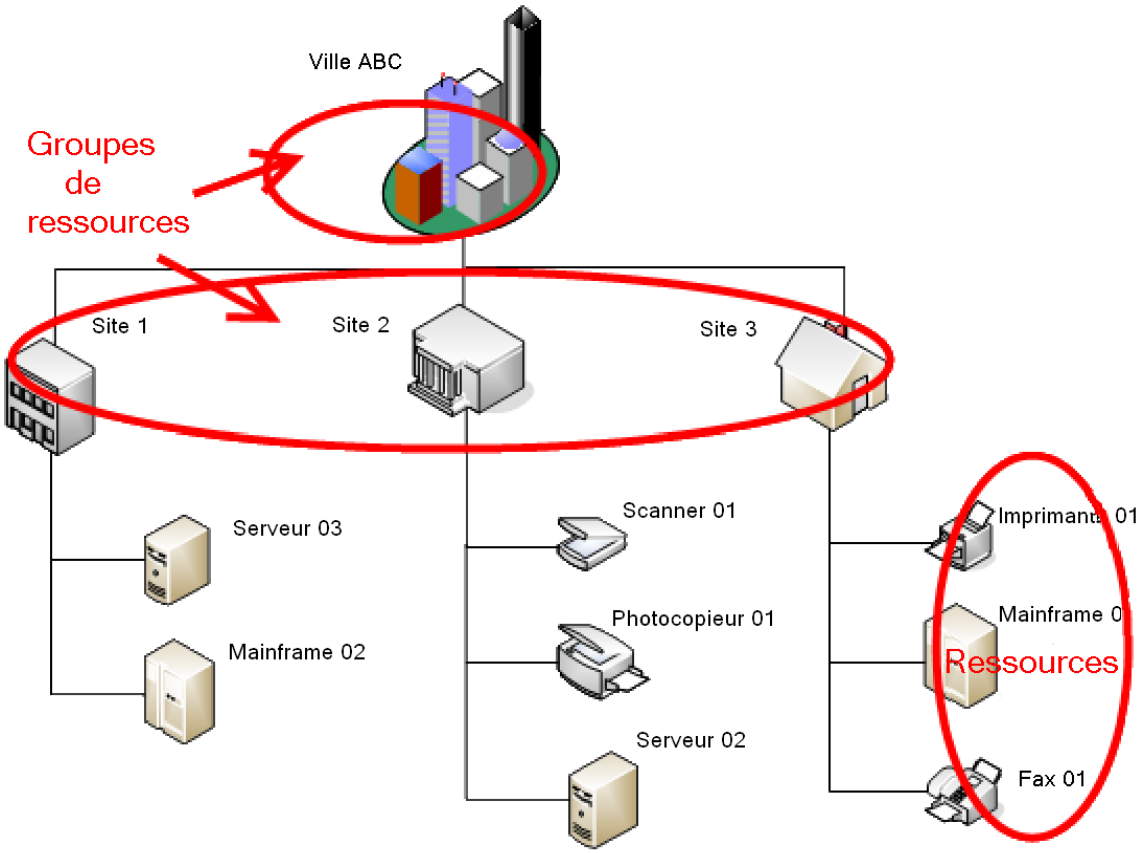
Dans les cas où cette méthode d'enregistrement est utilisée, la métrique calcule un résultat pour chacune des ressources dans le groupe de ressources où la métrique est regroupée,

ou

```
dispatcher.RegisterByResource "<ProcedureName>", "<Event Type name>",
"Context.ClusterItem
```

Dans les cas où cette méthode d'enregistrement est utilisée, la métrique calculera un résultat pour chacun des groupes de ressources contenus dans le groupe de ressources où la métrique est regroupée.

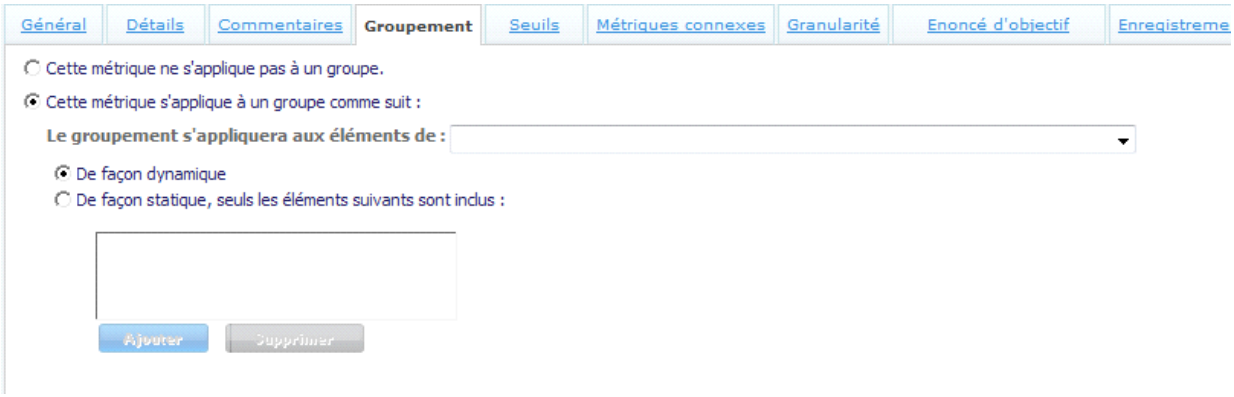
Le regroupement peut avoir lieu à des niveaux différents en fonction du mode de création du modèle de ressource. Le plus souvent, les organisations contiennent des couches différentes de groupement qu'elles souhaitent représenter. Par exemple, dans une ville particulière, il peut y avoir un nombre de sites, et dans chaque site il peut y avoir un nombre de périphériques d'infrastructure. (Imprimantes, analyseurs, serveurs etc) Grâce à ce type de regroupement, vous pourriez structurer une hiérarchie des ressources définie, qui contient plusieurs niveaux et regroupements de ces éléments de matériel. (En supposant qu'un périphérique d'infrastructure sera la "ressource". La structure décrite dans ce cas pourrait ressembler à ce qui suit :



Comme dans le diagramme, il y a plusieurs couches de groupes. Le groupe de niveau supérieur "Ville ABC" contient trois sites différents (qui sont également des groupes de ressources). Le groupe de ressources "Ressources du site 3" contient trois ressources différentes. Ainsi, à partir de l'exemple précédent, pour regrouper la métrique dans les trois sites différents, vous utiliseriez l'enregistrement suivant :

```
dispatcher.RegisterByResourceGroup  “<ProcedureName>”, “<Event Type name>”,
Context.ClusterItem
```

Dans ce cas, Context.ClusterItem fait référence au groupe de ressources appelé "Sites ABC de la ville", qui contient trois autres groupes de ressources appelés "Ressources du site 01", "Ressources du site 2", etc. et qui peuvent avoir cette configuration sur l'onglet Groupement de la métrique.



Remarquez également que le regroupement est défini sur dynamique car il inclura automatiquement tous les changements à venir apportés au groupe. Le groupement statique peut être utile pour des sous-ensembles de groupes de ressources ou lorsque vous ne souhaitez pas que le regroupement évolue dans le temps.

Pour créer une métrique qui génère des rapports sur les ressources du groupe Site 3, utilisez l'instruction d'enregistrement suivante :

```
dispatcher.RegisterByResource  “<ProcedureName>”, “<Event Type name>”,
Context.ClusterItem
```

Dans ce cas, Context.ClusterItem fait référence aux différentes ressources, d'où l'enregistrement par ressource uniquement. L'onglet Groupement de la métrique contient une référence au groupe Ressources du site 03.

Vous pouvez configurer le groupement pour qu'il fonctionne sur différents niveaux de la hiérarchie dans une métrique unique. Par exemple, en reprenant la situation décrite dans l'exemple précédent et le regroupement de cette métrique dans le groupe "Sites ABC de la ville". Vous pouvez inclure dans une même métrique les membres de ressource provenant de différents niveaux de la hiérarchie. Dans ce cas, il y a trois options affectant les ressources incluses dans ce groupement :

- **Premier niveau uniquement** : **Membres directs** : Identiques aux anciennes méthodes de regroupement décrites précédemment.
- **Tous les niveaux : Inclut des ressources uniquement** : Inclut toutes les ressources contenues dans les trois groupes de ressources de sites dans un seul niveau, et calcule le niveau de service pour chacun individuellement.
- **Tous les niveaux : Inclut des ressources et groupe de ressources** : Inclut toutes les ressources contenues dans les trois groupes de ressources de sites, ainsi que les trois groupes de ressources, et calcule le niveau de service pour chacun individuellement.

Agents

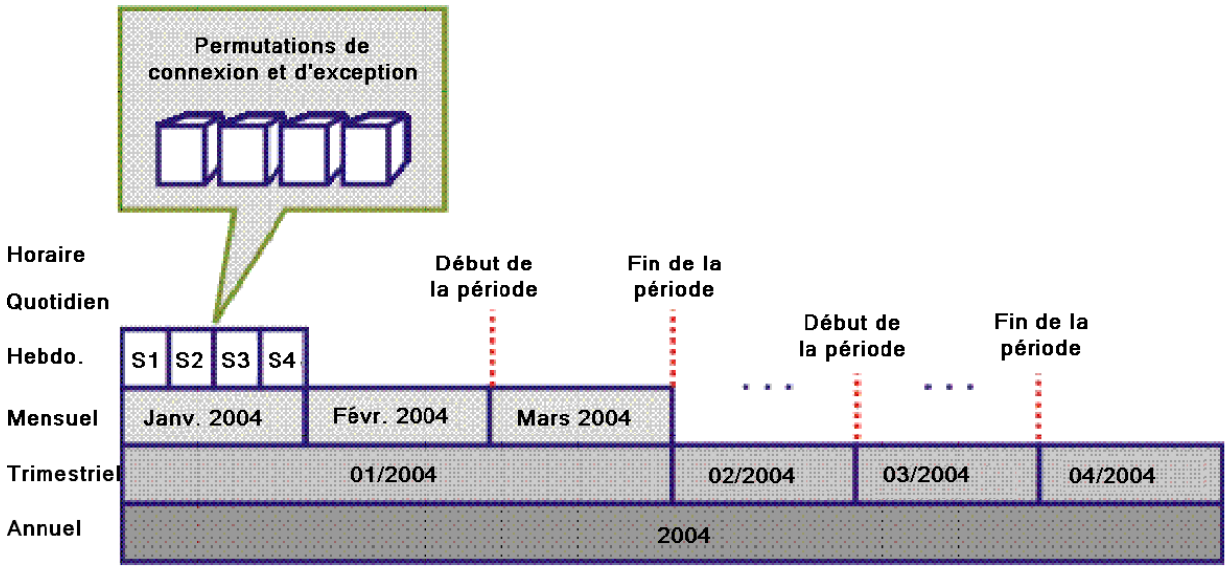
Chaque métrique contient une définition d'une période de suivi. La période de suivi est la période pendant laquelle la métrique doit produire un résultat de niveau de service et c'est ce résultat qui doit être comparé à la cible définie. Le résultat produit pendant la période de suivi est le résultat commercial, autrement dit, le résultat contractuel par lequel le fournisseur s'est engagé à fournir un certain niveau de service cible. Chaque instance de la logique applicative pour chaque période s'appelle un agent, et elle est directement liée aux granularités associées à chaque métrique.

Par exemple, si la métrique est définie avec une période de suivi d'un mois, alors la métrique est exécutée pour fournir un résultat mensuel.

Afin de fournir la fonction d'exploration vers le bas où l'utilisateur peut explorer vers le bas le résultat mensuel pour consulter le résultat quotidien, la métrique doit avoir une définition d'unités de temps supplémentaires dans lesquelles elle doit être calculée. Les unités de temps sont définies dans la section Détails généraux de la métrique dans l'onglet Granularité.

Pour chaque unité de temps définie dans l'onglet Granularité de la métrique et pour la période de suivi, une instance séparée de la logique applicative est exécutée par le moteur. Chacune de ces instances exécute le même code de logique applicative, mais OnPeriodStart et OnPeriodEnd seront déclenchées différemment pour chacune de ces instances. Pour l'instance quotidienne, celle-ci sera déclenchée au début et à la fin du jour. Pour chaque unité, elle sera déclenchée selon les points de démarrage et de fin de l'unité de temps.

Chacune des instances de la logique applicative est exécutée afin de produire le résultat d'unité de temps pertinent. De plus, chaque période doit avoir un résultat qui tient compte des exceptions. Une période qui ne fonctionne pas comme indiqué, (si les exceptions sont définies) doit permettre à l'utilisateur de choisir d'afficher le résultat de niveau de service avec ou sans les exceptions prises en considération. En raison de cette condition, chaque métrique aura potentiellement quatorze agents (instances) exécutés par le moteur, deux agents pour les résultats commerciaux et douze pour les six périodes opérationnelles supplémentaires.



Echelle de temps de logique applicative

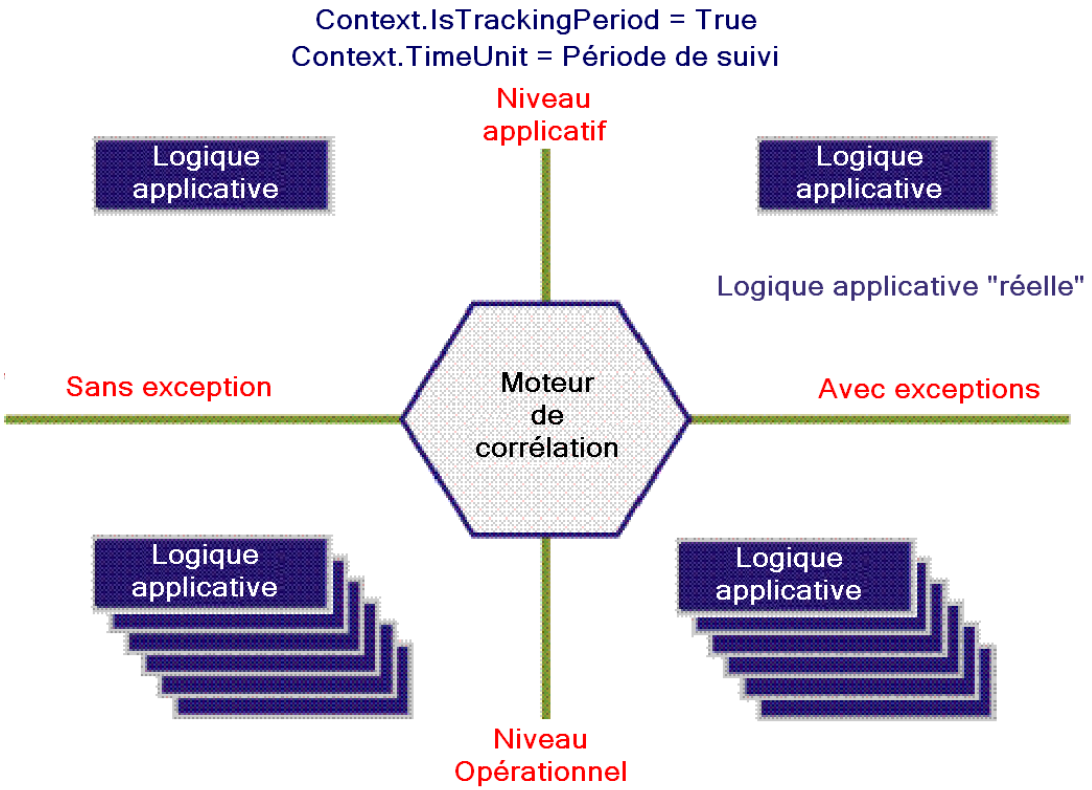
Cela veut dire que la définition de granularité a un impact important sur la performance du moteur de calcul parce que chaque période est calculée pour un agent différent. Dans les cas où la fonction d'exploration vers le bas n'est pas exigée complètement ou partiellement, il est donc recommandé de désactiver certains agents. Cette opération a un impact particulièrement grand pour les granularités inférieures comme les résultats horaires. Cette opération a également un impact important pour la métrique groupée, puisque le moteur effectue tous les calculs mentionnés ci-dessus pour chaque ClusterItem qu'il rencontre. Il traite chaque ClusterItem comme une nouvelle métrique. Par exemple, lors du calcul d'une métrique sur un groupe de ressources de 50 éléments, le moteur aura 49 fois plus de travail comparé à la même métrique non groupée.

Par exemple, si la métrique est définie pour calculer le temps de résolution en nombre de jours, alors un résultat horaire n'est pas pertinent et doit être désactivé sur l'onglet des granularités pour éviter au moteur d'accomplir des calculs inutiles.

L'attribut TimeUnit de l'objet Contexte (par ex. context.TimeUnit dans la logique applicative) renvoie l'unité de temps de l'agent actuellement en cours d'exécution où les valeurs renvoyées possibles sont : HEURE, JOUR, SEMAINE, MOIS, TRIMESTRE, ANNEE.

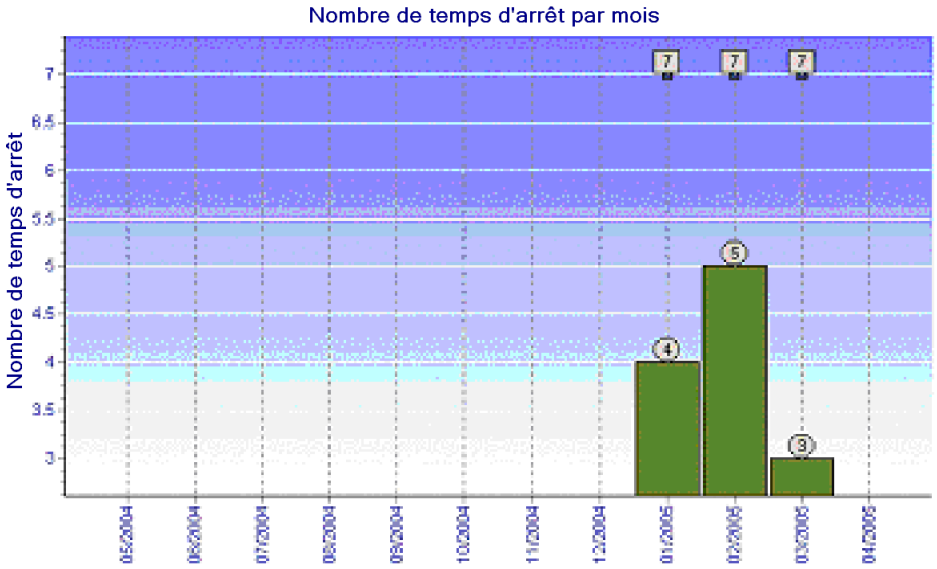
Par exemple, pour l'agent quotidien Context.TimeUnit renverra "JOUR".

L'attribut IsTrackingPeriod de l'objet Contexte renverra True pour l'agent qui calcule l'unité de temps de période de suivi. Il faut également noter que les agents affichés dans l'onglet Granularité des métriques viennent s'ajouter à l'agent de la période de suivi de la métrique. Ainsi, même pour une métrique avec une période de suivi mensuelle, vous pouvez désactiver l'agent mensuel qui calculera quand même le niveau de service mensuel, mais uniquement comme "Résultats commerciaux". (résultats non opérationnels)

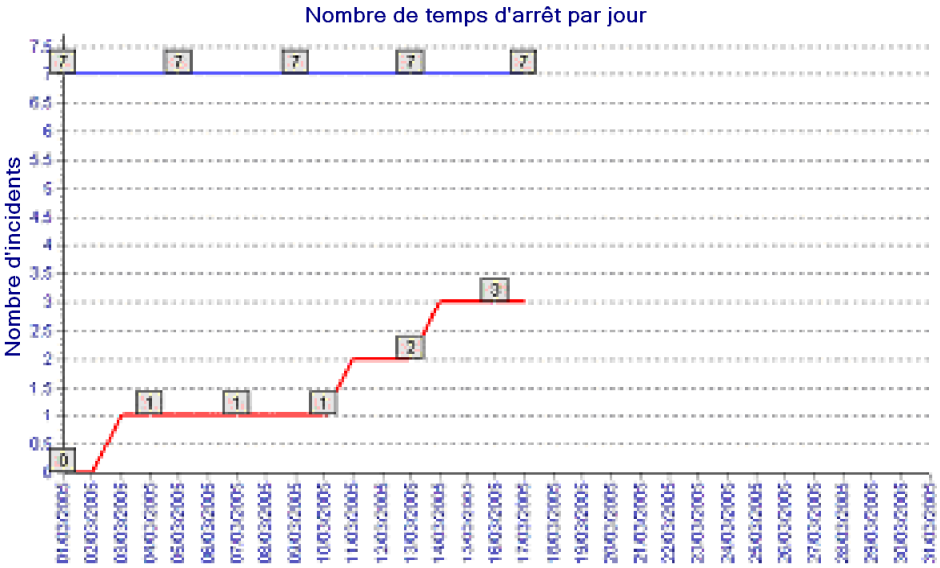

Context.IsTrackingPeriod = True
Context.TimeUnit = Période de suivi
Niveau applicatif
Logique applicative
Logique applicative
Logique applicative "réelle"
Sans exception
Moteur de corrélation
Avec exceptions
Logique applicative
Logique applicative
Niveau Opérationnel
Context.IsTrackingPeriod = False
Context.Time Unit="HOUR"I"DAY"I"WEEK"I"MONTH"I"QUARTER"I"YEAR"

Comme mentionné ci-dessus, les divers agents exécutent généralement le même code de la logique applicative mais il existe des cas où il est nécessaire d'appliquer une logique légèrement différente.

Par exemple, dans le cas mensuel, le résultat devrait être le nombre de fois pour chaque mois séparément comme indiqué ci-dessous :



Pour l'exploration quotidienne, il peut être nécessaire de consulter les cas cumulés où chaque jour est la somme de tous les jours à partir du début du mois. Cette méthode doit être appliquée à toutes les unités de temps inférieures à un seul mois, comme indiqué ci-dessous :



La différence entre les deux unités de temps est que pour l'agent qui calcule la période de suivi, le compte sera initialisé à 0 au début de chaque période, mais pour l'agent quotidien, le décompte sera initialisé uniquement au cas où le jour est le premier jour du mois.

```
Les informations suivantes sont le code de gestionnaire d'événements
OnPeriodStart :
Sub OnPeriodStart(time)
     If InStr('MONTH,QUARTER,YEAR', Context.TimeUnit) > 0 _
 Or (Context.TimeUnit = 'DAY' And Day(time) = 1) _
 Or (Context.TimeUnit = 'HOUR' And Day(time) = 1 And Hour(time) = 0) Then
          DownTimeCounter = 0
     End If
End sub
```

Qu'est-ce que le nouveau calcul ?

Le nouveau calcul s'accomplit lorsque le moteur de corrélation identifie que le résultat périodique final d'une métrique n'est plus valide et qu'il doit par conséquent recalculer les résultats.

Le nouveau calcul peut être causé par :

- La réception de nouveaux événements qui se sont produits dans le passé. (antérieurs au *calcul effectué jusqu'à présent par le moteur* pour cette métrique)
- Une ressource enregistrée dans la métrique est modifiée (nouvelle date de version et changements connexes).
- La validation d'une nouvelle version de contrat qui fait se chevaucher l'heure du précédent calcul avec les changements affectant certaines métriques (seules les métriques modifiées sont recalculées)

La méthode la plus efficace pour l'enregistrement est l'utilisation de contractant et de service. Organiser les ressources sous contractant et service est un moyen d'exprimer la relation logique entre la couche de données et la couche d'activité professionnelle dans le système. Le fait d'enregistrer les ressources via ces entités nécessite qu'aucun changement ne soit apporté aux formules dans le cadre d'une utilisation dans des contrats différents ou lorsqu'elles sont utilisées pour des services différents. Le contexte actuel de la métrique a identifié le contrat et le service, ce qui définit le contractant adéquat et le service associé. Les formules de la logique applicative qui sont définies dans ce type d'enregistrement sont facilement réutilisables parce qu'elles n'ont pas besoin des changements dans l'enregistrement.

Sorties : Tables utilisateur

Le script de la logique applicative standard n'a pas accès aux tables de sortie externes. Il n'existe que deux destinations de sortie :

- La table PSL (T_PSL) dans laquelle le moteur écrit automatiquement les résultats du niveau de service ainsi que le résultat spécifié dans la fonction de résultat. Les valeurs de niveau de service qui sont écrites dans cette table peuvent être seulement d'un type numérique. Les résultats écrits dans la table T_PSL sont les résultats retournés par l'assistant de création de rapports. Il n'y a pas de contrôle sur la structure ou la méthode dans lesquelles sont écrits ces résultats. Cette opération est effectuée automatiquement par le moteur de corrélation.
- La table de sortie de SLALOM (T_SLALOM_OUTPUTS). L'écriture dans cette table s'effectue à l'aide des méthodes fournies par l'objet Tools à partir de la formule de la logique applicative. Lors de la sortie dans cette table, un nom de table logique est fourni référencé sous le nom de table utilisateur. Ces tables servent à sortir des informations pendant le calcul au niveau du service. Vous pouvez plus tard utiliser ces informations pour générer des rapports en format libre. Il peut y avoir plusieurs tables utilisateur.

L'utilisation de la table externe T_SLALOM_OUTPUTS est requise lorsqu'une sortie supplémentaire est nécessaire sur et au-dessus du résultat du niveau de service périodique, lorsque vous ne pouvez pas fournir la sortie supplémentaire en ajoutant une autre métrique, ou lorsque l'ajout d'une autre métrique diminue la performance du calcul en traversant le même jeu d'enregistrements pour produire seulement une autre sortie.

Par exemple, considérez un cas où une métrique est définie pour calculer le pourcentage de tickets ayant été résolus en moins d'une journée et un rapport à produire contenant la liste de tous les tickets dont la résolution a échoué en moins d'un jour, il est nécessaire que la formule produise dans une table externe chaque ticket identifié comme défaillant et l'ajoute aux statistiques de calcul.

Dans le cadre de la condition ci-dessus, la table de niveau de service de sortie normale ne peut pas fournir cette sortie, parce que :

- Les résultats de service sont tous numériques
- Seul un résultat de niveau de service unique est possible pour chaque période.

Les enregistrements sont écrits dans des tables de sortie utilisateur uniquement pour l'agent qui s'exécute pendant la période de suivi de la métrique et calcule les exceptions et les corrections. Par exemple, si la métrique est définie comme ayant une période de suivi mensuelle, les instructions de sortie (Tools.SaveRecord et Tools.SaveFields) ne sont PAS exécutées lorsque le moteur exécute la formule pour les autres granularités comme l'HEURE, le JOUR, la SEMAINE, le TRIMESTRE et l'ANNEE

L'avantage supplémentaire de cette table externe est qu'elle peut satisfaire à plusieurs exigences de génération de rapports. Vous pouvez créer d'autres exigences de génération de rapports typiques à partir de ces tables en plus des exigences contractuelles. Par exemple, une métrique calculant le nombre de tickets clôturés en un mois pourrait également calculer le temps de résolution de ticket et produire la sortie de l'ensemble de ces informations dans la table de sortie. Cela permettra une analyse plus détaillée des tickets individuels qui ont été clôturés pendant la période ainsi que les détails supplémentaires nécessaires dans le cadre d'une exigence de génération de rapports distincts.

Les informations dans ces tables sont également gérées par le mécanisme de nouveau calcul du moteur. Pendant ce processus de nouveau calcul, les enregistrements sont marqués comme inactifs, et une nouvelle ligne est écrite à la place. Il s'agit d'un point important puisque que toutes les instructions SQL de rapport en format libre doivent inclure une vérification du champ IS_ACTIVE dans la table de T_SLALOM_OUTPUTS (où is_active = 1) puisque seuls ces enregistrements sont actuels.

Remarque : Lors de l'exécution de la portée de la logique applicative pendant le processus de débogage des formules, les messages sont en fait écrits dans la table T_DEBUG_SLALOM_OUTPUTS au lieu de la table T_SLALOM_OUTPUTS.

Lors de la documentation des données à l'aide de T_SLALOM_OUTPUTS, les données insérées sont toujours du texte (puisque les champs de T_SLALOM_OUTPUTS sont tous varchar2). Par conséquent, les valeurs de date sont transformées en texte en appliquant le format du système d'exploitation qui peut changer pendant le cycle de vie de l'application. T_SLALOM_OUTPUTS peut par conséquent connaître des incohérences dans les formats de date. De plus, la logique applicative gère des dates UTC où il est possible que T_SLALOM_OUTPUTS contienne des horodatages locaux, ainsi dans certains cas il peut être nécessaire d'utiliser la fonction de conversion *Tools.GetLocaleDate(date)* pour résoudre le problème.

```
La fonction suivante transforme des dates en heures locales et conserve la
cohérence du format de date en convertissant les dates au format "jj/mm/aaaa
hh24:mi:ss" :
Function FormatDate(time)
     Dim LocalTime
```

```
    LocalTime=Tools.GetLocaleTime(time)
    FormatDate=Day(LocalTime) & "/" & Month(LocalTime) & "/" &
  Year(LocalTime) & " " & _
    Hour(LocalTime) & ":" & Minute(LocalTime) & ":" & Second(LocalTime)
End Function
```

Vous pouvez faire appel aux deux méthodes suivantes pour écrire dans la table externe depuis la formule de logique applicative :

- Tools.SaveRecord <parameter list>
- Tools.SaveFields <parameter list>

Ces deux méthodes de l'objet Tools sont décrites en détail ci-dessous :

```
Tools.SaveRecord tableName, key,[val1],[val2],…
```

Cette méthode sauvegarde un enregistrement dans une table appelée T_SLALOM_OUTPUTS. Le paramètre tableName spécifie la table (virtuelle) dans T_SLALOM_OUTPUTS dans laquelle les informations devraient être écrites. Chaque enregistrement dans la table d'utilisateur a une clé unique qui spécifie l'enregistrement dans lequel les informations devraient être écrites. Chaque enregistrement contient également jusqu'à 20 champs de valeur de type chaîne. La méthode SaveRecord reçoit le nom de la table utilisateur et une clé. Elle accepte également tous les champs de valeur dans la table utilisateur. (Ces paramètres de valeur sont facultatifs et peuvent être ignorés.) Si un enregistrement avec la même clé existe déjà, il est mis à jour. (Seuls les champs de valeur transférés en tant que paramètres sont mis à jour.) Si aucun enregistrement possédant cette clé n'existe, il est créé.

```
Tools.SaveFields tableName, key, [fieldName1,fieldVal1], [fieldName2,fieldVal2]
```

Cette méthode est similaire à SaveRecord, sauf qu'au lieu d'énumérer toutes les valeurs, elle fournit des paires de noms de champ et des valeurs de champ connexes. Vous pouvez remplacer les numéros de champ par les noms de champ. Les noms de champ doivent auparavant être manuellement définis dans la table T_SO_FIELD_NAMES. Cette table est utilisée pour enregistrer la structure des tables de sortie.

Il est recommandé que l'expert en logique applicative définisse la structure de la table de sortie avant d'écrire dans T_SLALOM_OUTPUTS car la structure et la signification des champs sont déjà correctement définies et simplifieront nettement la requête.

La structure de table est :

- **table_name :** Chaque table logique possède un nom unique
- **field_name** : Chaque champ dans une table est unique

- **field_id :** Chaque champ porte un numéro de série commençant à partir de 1

Il est préférable d'utiliser la méthode SaveFields car elle garde une documentation de la structure et la signification des valeurs insérées.

***Exemple* :**

Considérez un cas où il est nécessaire de produire une liste de tous les incidents avec le temps de résolution plus élevé qu'un seuil défini, en plus du résultat de la métrique chargée de calculer le pourcentage de tickets qui ont été résolus en dessous ce seuil. L'écriture dans les tables de sortie se fera dans la procédure du gestionnaire d'événements OnXXXEvent après la validation de seuil.

Cette opération est illustrée par l'exemple suivant :

```
Sub OnIncidentEvent(eventDetails)

    If eventDetails("RESOLUTION_TIME")<=SThreshold Then
     CountSuccessIncident s= CountSuccessIncidents+1
    Else
création de la clé unique d'enregistrement
     KeyString= eventDetails("IncidentID") &eventDetails.Time

        Tools.SaveFields "IncidentsTable", KeyString, "CASE_ID",
    eventDetails("CASE_ID"),_
           "CUSTOMER_REF",eventDetails("CUSTOMER_REF"),_
           "TICKET_CREATOR",eventDetails("TICKET_CREATOR"),_
           "CREATION_TIME",eventDetails("CREATION_TIME"),_
           "SEVERITY",eventDetails("SEVERITY"),_
           "RESOLUTION_TIME",eventDetails("RESOLUTION_TIME"),_
           "CLOSE_TIME",eventDetails("CLOSE_TIME")
    End If
End sub
```

Les éléments suivants sont des propositions associées à l'utilisation des tables T_SLALOM_OUTPUTS :

- Il est recommandé d'écrire uniquement les données nécessaires dans les tables de sortie (pas plus).
- Ecrivez dans les tables de sortie uniquement les granularités de métrique qui sont nécessaires.
- Tous les champs de valeur ne contiennent que 50 caractères par défaut (sauf la colonne VAL_1, qui en contient 512). Vous pouvez avoir besoin de tronquer certaines valeurs pour les champs afin d'ajuster la taille de données et d'éliminer les erreurs d'exécution dans la logique applicative.
- Il n'y a que 20 colonnes disponibles pour vos données (VAL_1... VAL_20)

Remarque : Ecrire dans les tables de sortie peut avoir une influence sur la performance du moteur, car l'écriture dans une table est intensive sur le plan calculatoire par comparaison à un calcul en mémoire. Par conséquent réfléchissez s'il est nécessaire d'utiliser cette fonctionnalité et à quel emplacement elle est requise, et réduisez le nombre des accès aux tables.

Rapport sur les informations issues des tables définies par l'utilisateur

Il n'est pas possible d'utiliser l'assistant de création de rapports CA Business Service Insight pour générer un rapport sur les informations écrites dans les tables de sortie. Afin de créer un rapport sur ces informations, il est nécessaire de créer un rapport au format libre. Il s'agit essentiellement de créer une requête SQL au-dessus de cette table. Comme cette table contient beaucoup de tables logiques qui font l'objet d'écriture par diverses formules, il est recommandé qu'un expert SQL (expert en sources de données) crée une vue sur T_SLALOM_OUTPUTS afin de mieux faire la différence entre les différentes tables logiques contenues dans celle-ci et de s'assurer également que les informations sont récupérées comme prévu.

La définition de vue aura déjà toute la diffusion des types de champs de chaîne en type d'informations pertinent et contiendra également l'ensemble des filtres nécessaires (table logique, enregistrements actifs, métrique pertinente, etc.).

Les éléments suivants sont un exemple de création de vue :

```
Créez ou remplacez la vue kpi_view comme
select
     to_date(t...) comme fieldName,
     to_number(t..)...
de t_slalom_outputs t,
     t_rules r,
     t_sla_versions sv,
     t_slas s,
où table_name = 'TableName'
     et is_active = 1
     et t.rule_id = r.psl_rule_id
     et r.sla_version_id = sv.sla_version_id
     et sv.sla_id = s.sla_id
     et sv.status = 'EFFECTIVE'
```

Création des modules de logique applicative

Les utilisateurs peuvent définir des modules de logique applicative indépendants qui peuvent être utilisés par des objectifs de niveau de service multiples (métriques). Pour appliquer des changements importants à la logique applicative dans l'ensemble du système, l'utilisateur modifie le composant de la logique "de base" qui peut ensuite être appliqué d'un seul clic à tous les SLA pertinents.

Un module de logique applicative est un composant de code dans lequel la logique applicative peut être définie et maintenue facilement, et qui réduit la redondance. Un module unique peut être utilisé par plusieurs métriques.

Pendant l'étape de configuration, les formules sont configurées pour définir les modules de logique applicative principaux. (Voir le chapitre Conception : Modèles et modules de logique applicative.)

Il existe trois types de modules de logique applicative :

- **Logique applicative complète** : Module utilisé lorsqu'une formule entière doit être utilisée comme module. Les méthodes Result et OnRegistration doivent être implémentées dans le script de module.
- **Classe** : Module qui contient une définition d'une classe de VB unique.
- **Bibliothèque** : Module qui contient une collection de procédures, de fonctions et de classes.

Vous pouvez utiliser des modules avec les éléments suivants :

- Métriques
- Autres modules
- Modèles de logique applicative
- Métriques dans des modèles de contrat et des modèles de niveau de service.

Les modules peuvent utiliser des paramètres qui sont extraits des paramètres du contexte de métrique ("ParamName").

Remarque : Pour éviter des erreurs d'exécution, définissez toujours une valeur par défaut lors de l'utilisation des paramètres dans des modules de logique applicative. La formule produit un message d'erreur de journal pour des paramètres non existants.

```
If Parameters.IsExist("ReportedDowntimesNum") Then
    maxBufferSize = Parameters("ReportedDowntimesNum")
Else
```

```
    maxBufferSize = 3
    out.log "ReportedDowntimesNum parameter not set", "E"
End If
```

Exemple de module de logique applicative

Il existe un objet de niveau de service décrit comme "assistance suivant des limites définies." Le système d'assistance suivant repose sur le cycle de vie des tickets ayant pour état :

- Ouvert
- Affecté
- Clôturé.

Les deux métriques que vous pouvez définir afin de décrire la performance de l'assistance sont :

- **Nom de la métrique** : **Résolution de ticket réussie dans les temp**s.

  Enoncé d'objectif : Au moins 99% des tickets doivent être résolus dans les 4 heures.

  Logique applicative : La résolution doit être calculée entre Ouvert et Clôturé.

- **Nom de la métrique** : **Affectation de ticket réussie dans les temps**.

  Enoncé d'objectif : Au moins 99% des tickets doivent être résolus dans les 30 heures.

  Logique applicative : L'affectation doit être calculée entre Ouvert et Clôturé.

Comme vous pouvez identifier la même logique pour ces deux métriques, vous pouvez créer un module pour s'adresser aux deux métriques.

Le module exige les paramètres suivants dans le contexte de la métrique :

- Premier statut
- Deuxième statut
- Seuil.

Une fois que le module de logique applicative a été créé, les métriques définies peuvent utiliser un module dans le cadre de la définition.

Vous pouvez ensuite modifier la logique. Vous pouvez, par exemple, considérer un nouveau statut Client en attente. Ce statut est défini pour un ticket lorsque l'assistance attend des informations supplémentaires relatives au ticket du client. Dans la logique applicative, lorsque le ticket est en mode Client en attente, il ne doit pas être considéré comme appartenant au calcul. Par conséquent, seul le module de logique applicative doit changer pour tenir compte du nouveau statut et de la logique. Une nouvelle version du module, y compris la nouvelle logique, est créée.

Lors de la validation de la modification, vous voyez apparaître une liste des métriques utilisant le module de logique applicative. Vous pouvez soit appliquer la modification à toutes les métriques collectivement, soit choisir d'appliquer la modification uniquement aux métriques spécifiques dans la liste.

Si vous sélectionnez uniquement des métriques spécifiques dans la liste, vous êtes invité à créer un nouveau module pour les métriques sélectionnées. Le précédent module de logique applicative utilisé par les métriques sélectionnées est remplacé par le nouveau module de logique applicative et le calcul est réexécuté en fonction de la nouvelle métrique.

Création des paramètres

Les paramètres fournissent aux utilisateurs professionnels une façon rapide et intuitive d'afficher et de changer leurs valeurs en utilisant l'interface utilisateur graphique sans devoir modifier le script de formule.

L'utilisation de paramètres dans la logique applicative permet la création de formules générales qui peuvent avoir une utilisation importante dans le système, ce qui optimise l'utilisation des modules.

Vous pouvez définir des paramètres au niveau du contrat ou de la métrique. Les paramètres de métrique sont affichés et configurés dans l'onglet Enoncé d'objectif des détails de la métrique. L'accès de la logique applicative est limité aux paramètres au niveau de la métrique ; par conséquent, afin d'accéder à un paramètre de contrat dans une métrique, un type différent de paramètre est créé localement dans la métrique. Il s'agit d'un paramètre dynamique qui prend sa valeur de référence à partir des paramètres au niveau du contrat. Les valeurs de référence autorisées dans le paramètre dynamique sont uniquement celles définies dans le contrat parent de la métrique.

Types de paramètres :

- Date : En mode date (Date + Heure).
- Nombre : Taille maximale des 15 chiffres avec une précision maximale de 5 chiffres.
- Texte : Taille maximale de texte de 100 caractères.
- Table : Taille maximale de table de 120 entrées.

Afin d'accéder aux valeurs du paramètre à partir du code de formule, il est nécessaire d'utiliser l'objet Paramètres et de se référer au nom du paramètre.

Exemple **:**

```
Paramètres ("Threshold")
(Notez qu'il s'agit d'une méthode abrégée pour appeler la valeur ; normalement
cette opération s'effectue avec Parameters.Item("Threshold"))
```

Ou, pour un paramètre de type table :

Paramètres ("Table")("Word")("Price")

(où les valeurs "Word" et "Price" sont les noms de ligne et de colonne du paramètre de table, respectivement)

Les paramètres de table doivent uniquement être utilisés selon un nombre de points clés :

1. Définir une variable globale (par ex. dim myTableParam)
2. Dans la fonction OnLoad, remplissez la variable depuis l'objet de paramètres (ex. “Set myTableParam = Parameters(“MyParametersTable”))
3. Par la suite utilisez uniquement l'objet créé, myTableParam.
   Le paramètre ne doit pas être utilisé en dehors de la fonction OnLoad et vous devez uniquement vous référer à l'objet variable global que vous avez créé à partir de celui-ci.(par ex. “dim myVal: myVal = myTableParam (“myRow”)(“myColumn”)).

Cette opération libère la mémoire que le paramètre utilise et empêche le moteur de créer le mappage du paramètre sur chaque appel de paramètre et pour chaque "OnXXXEvent" que vous pouvez appeler des milliers de fois par métrique

Le code suivant illustre l'utilisation correcte d'un paramètre de table :

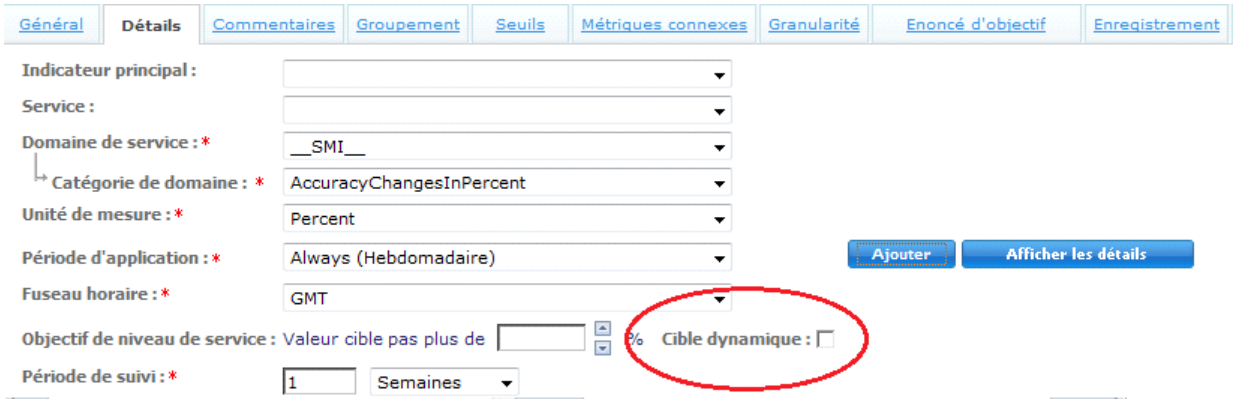
```
Option Explicit

Dim sum
Dim myParamTable

Sub OnLoad(TIME)
   Set myParamTable = Parameters("MyTableParam")
End sub

Sub OnRegistration(dispatcher)
  dispatcher.RegisterByResource" OnEvent", "EventType", "ResourceType"
End sub

Sub OnPeriodStart(TIME)
   sum = 0
End sub

Sub OnEvent(eventDetails)
   If Context.IsWithinTimeslot Then
      sum = sum + 1 * myParTimeSlotamTable("firstRow")("secondColumn")
   End If
End sub

Function Result
   Result = ( sum * myParamTable("secondRow")("thirdColumn")  )
End Function
```

Les méthodes suivantes sont disponibles pour chaque objet de paramètre créé dans le code.

Paramètre	Description
IsExist	Paramètre existant.
IsNumeric	Paramètre de type numérique.
IsText	Paramètre de type texte.
IsDate	Paramètre de type date.
IsTable	Paramètre de type table.
IsEntryExist	Entrée dans la table.
IsEntryNumeric	Entrée dans la table de type numérique.
IsEntryText	Entrée dans la table de type texte.
IsEntryDate	Entrée dans la table de type date.
Dump	Renvoie une liste de tous les paramètres.
Item	Fait référence à la valeur de paramètre.

Implémentation des cibles dynamiques

Les cibles dynamiques sont gérées par la logique applicative à l'aide d'un gestionnaire d'événements dans le script de la logique applicative standard, similaire à la fonction Result() qui permet de retourner la valeur de niveau de service à partir de la métrique. La cible dynamique doit être spécifiée sur l'onglet Détails de la métrique comme indiqué ci-dessous.



Lorsqu'une cible dynamique est spécifiée, la cible est extraite de la fonction "Target()" dans la logique applicative plutôt que dans la valeur statique spécifiée dans l'onglet Détails de la métrique. La fonction Target ressemble à ce qui suit.

```
Function Target
  'TODO : Ajoutez du code ici pour traiter le calcul cible dynamique
  Target = Null
End Function
```

Cette fonction doit être implémentée à partir de la condition pour la métrique afin de renvoyer la valeur cible désirée pour une période spécifique. La fonction peut renvoyer toute valeur que la logique applicative peut lui affecter.

Exemple réel de cibles dynamiques

Pour un centre d'appel, la cible d'une métrique évaluant *Avg Call Pickup time* peut dépendre du volume des appels. S'il y a entre 0 et 800 appels, la cible doit être inférieure à 15 secondes, entre 801 et 1500 appels, la cible doit être inférieure à 20 secondes, plus de 1500 appels, la cible doit être inférieure à 25 secondes. Vous pouvez l'implémenter comme suit : (en supposant que TotalCalls est un compteur incrémenté à chaque événement d'appel reçu et que TotalCalls ne peut pas être inférieur à 0)

```
Function Target
  If TotalCalls >0 et TotalCalls <= 800 Then
     Cible = 15
  ElseIf Total Calls > 800 et TotalCalls <= 1500 Then
     Cible = 20
  Else
      Cible = 25
  End If
End Function
```

Un autre exemple d'utilisation d'une cible dynamique :

Considérez la situation où la cible d'une métrique peut changer selon la granularité du calcul. Il est possible qu'il y ait une cible quotidienne de 98% de disponibilité pour un groupe de serveurs, mais la cible mensuelle est 99,5% de disponibilité. La solution dans ce cas nécessite d'utiliser la fonction cible dynamique conjointement à l'appel de fonction à *Context.TimeUnit* pour déterminer l'agent actuel en cours de calcul. Vous pouvez donc ajuster la cible en conséquence.

```
Function Target
  If Context.TimeUnit = 'DAY' Then
     Cible = 98
  ElseIf
     Cible = 99,5
  End If
End Function
```

Sauvegarde des états

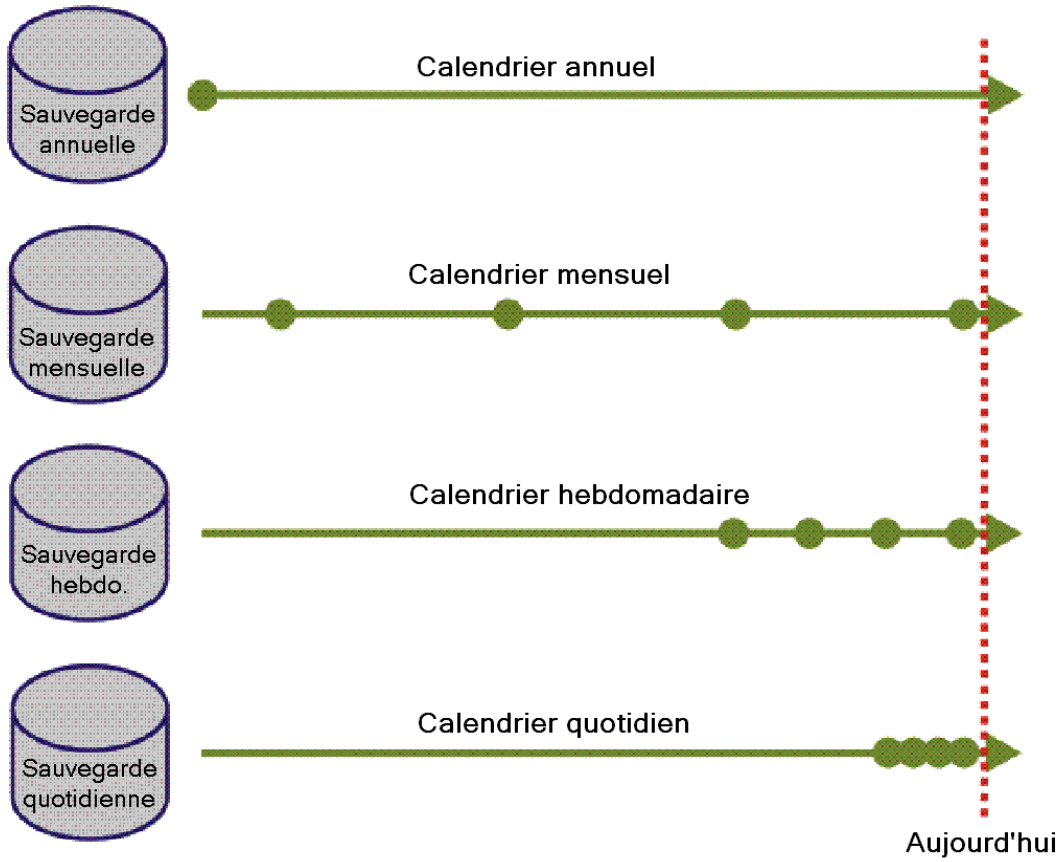
Pendant le processus continuel de calcul des niveaux de service pour chaque métrique, le moteur est souvent obligé d'accomplir un calcul partiel pendant une période qui ne s'est pas encore terminée. Pour éviter un retour au début du calcul lorsque des nouvelles données arrivent progressivement, le moteur accomplit un type de sauvegarde de son "état" en cours avant de passer à sa tâche de calcul suivante. A ce stade, il effectue un instantané des variables et des valeurs actuelles à ce stade du calcul et enregistre cet "état" dans la base de données.

Le processus de sauvegarde de la logique applicative est un mécanisme selon lequel le code de la logique applicative, y compris les valeurs des variables, est codé dans un flux binaire et enregistré dans la base de données. Ce mécanisme est également nécessaire pour accélérer la performance du moteur de calcul pour les nouveaux calculs. L'état est sauvegardé de manière sporadique, et est utilisé dans le nouveau calcul ainsi que comme mesure de l'efficacité pour des calculs en continu.

Par exemple, si un nouveau calcul est nécessaire pour un mois de manière rétroactive, au lieu de recalculer les résultats depuis le début du contrat, l'état sauvegardé le plus proche avant la date de nouveau calcul est utilisé et les calculs sont accomplis à partir de cet état.

Le moteur de calcul utilise une méthode heuristique prédéfinie pour déterminer à quel moment la sauvegarde est nécessaire et utilise les fonctionnalités de sauvegarde pour stocker l'état codé dans la base de données.

Dans l'illustration suivante, les points rouges représentent une sauvegarde d'état. Plus la considération remonte, moins il y a d'états sauvegardés pris en compte. La logique de ce mécanisme repose sur la supposition que le nouveau calcul est habituellement nécessaire pour la période qui remonte avant un mois.


Sauvegarde annuelle
Calendrier annuel
Sauvegarde mensuelle
Calendrier mensuel
Sauvegarde hebdo.
Calendrier hebdomadaire
Sauvegarde quotidienne
Calendrier quotidien
Aujourd'hui

Optimisation du système pour le nouveau calcul

Le processus de calcul de niveau de service exige une quantité considérable de ressource UC, de mémoire et de stockage. Ce qui suit est une liste de recommandations qui peuvent réduire la charge informatique et améliorer la vitesse de calcul.

Remarque : Certaines recommandations exigent des paramètres internes qui ne sont pas disponibles dans l'interface utilisateur. Dans ces cas, contactez le support technique de CA pour plus de détails et d'instructions.

- **Modifiez la configuration d'enregistrement des états**

  Selon les retards connus de l'adaptateur, vous pouvez définir les paramètres des états pour mieux répondre à votre configuration. Autrement dit, vous pouvez changer le nombre et la granularité des points d'enregistrement des états.

- **Configurez le système pour calculer uniquement les unités de temps qui sont réellement nécessaires**

  Les métriques peuvent contenir jusqu'à sept périodes de granularités différentes : *année*, *quart*, *mois*, *semaine*, *jour*, *heure*, et la période de suivi. Dans certaines implémentations, toutes les granularités ne sont pas nécessaires. Vous pouvez désactiver les granularités inutiles pour les contrats non validés via l'interface utilisateur. Reportez-vous à chaque onglet de granularité des métriques. Pour changer les granularités de la métrique de contrats validés, contactez le support technique de CA.

  Remarque : Evitez le calcul de périodes opérationnelles qui sont similaires à la période de suivi.

- **Ne calculez pas les valeurs "sans correction", "sans exceptions"**

  Valeur par défaut, le moteur de calcul calcule quatre valeurs différentes : *fourni*, *fourni avec corrections*, *fourni avec exceptions* et *fourni avec corrections et exceptions*. Vous pouvez changer cela pour calculer uniquement la valeur fournie.

  Remarque : Pour plus d'informations, contactez le support technique de CA.

- **Modifiez l'ordre de calcul**

  Vous pouvez définir des priorités pour l'ordre de calcul PSL des métriques pour commencer par vos métriques les plus importantes, puis passer aux autres.

  Remarque : Pour plus d'informations, contactez le support technique de CA.

- **Créez plusieurs instances PslWriter**

En créant plusieurs instances PSL (du moteur de calcul), vous pouvez fractionner la charge de travail et augmenter la vitesse de calcul.

Remarque : Pour plus d'informations, contactez le support technique de CA.

- **Prévoyez des implémentations pour raccourcir le temps de nouveau calcul**

  Pour optimiser le temps de nouveau calcul, vous pouvez :

  - Exécuter l'adaptateur plus souvent afin de réduire les événements retardés et éviter des accumulations importantes de données à traiter par le moteur
  - Désactiver des agents inutilisés dans l'onglet de granularité des métriques
  - Dupliquer les métriques et calculer des agents différents à l'aide de la même métrique pour équilibrer la charge de calcul
  - Utiliser des métriques intermédiaires pour accomplir des calculs courants et pour partager les résultats avec toutes les autres métriques qui nécessitent les mêmes données.

- **Prévoyez des implémentations pour réduire la quantité de données**

  Pour optimiser la quantité de données, utilisez l'adaptateur pour charger uniquement les données rassemblées (traitées). Rassembler les informations de source de données avant de les envoyer à la table de données brutes CA Business Service Insight augmente l'efficacité de lecture d'entrée de PSL.

- **Suivez les recommandations de configurations CA Business Service Insight PSL**

  Vous pouvez reconfigurer le moteur PSL pour une meilleure performance en fonction de l'environnement de l'application et des exigences spécifiques. Vous pouvez définir certains paramètres de l'interface utilisateur et vous pouvez en définir d'autres uniquement depuis la table de configuration du système de base de données.

  Remarque : Consultez les recommandations de prise en charge pour vos configurations PSL.

Historiques et alertes

Il y a des cas où la logique applicative est requise pour créer un rapport dans le journal ou déclencher un message alerte. Cela est nécessaire lorsque les messages doivent être envoyés en fonction du traitement d'événement. Vous pouvez envoyer toutes les informations qui sont rassemblées pendant le processus du calcul et qui peuvent être pertinentes comme alerte. Par exemple, vous pouvez envoyer un message d'alerte lorsqu'un événement spécifique est sous le seuil horaire de résolution spécifié ou, dans le cadre d'une analyse de tendance, lorsqu'un certain nombre de défaillances consécutives a été atteint.

"Out" est un objet de la logique applicative globale qui permet à la formule d'envoyer des alertes ainsi que des messages de journal. Il contient deux méthodes associées qui ont la forme suivante :

```
Alert(<Event type>, <Resource name>, <value1, value2>, …<value16>)
```

Cette commande envoie une alerte d'un type d'événement spécifié. Toutefois ce type d'événement doit être créé manuellement pour cette alerte. Le nombre de valeurs et leur type doivent correspondre à la définition du type d'événement.

```
Log(<Message>,<Level>)
```

Cette commande envoie un message au journal du système. Le premier paramètre est le message d'informations signalé, et peut être du texte libre. Vous pouvez également ajouter les valeurs de variables à cette chaîne pour donner une signification contextuelle au message. Le paramètre "Level" peut prendre une des valeurs suivantes :

	Description
	Un message d'avertissement est signalé.
	Un message d'erreur est signalé.
	Un message d'information est signalé uniquement lors de l'exécution dans la portée de la logique applicative. Lors de l'exécution dans PslWriter, aucun message n'est signalé. Il s'agit du paramètre par défaut. Il est utilisé surtout à des fins de débogage.

Exemple **:**

Les informations suivantes sont extraites d'un cas où les informations d'infrastructure de l'événement étaient attendues avant les détails d'incident réels. Un mécanisme d'alerte a été configuré pour notifier l'administrateur de cette condition afin de demander la correction du problème.

```
Out.Alert "Site Unknown Alert", Context.ClusterItem, Context.Rule

Out.Log("Fault Event Received for a Site with no infrastructure details: " &
Context.ClusterItem)
```

Ajoutez des commentaires sur les causes initiales des problèmes et des commentaires d'événement

Vous pouvez définir des commentaires sur les causes initiales des problèmes dans la logique applicative pour expliquer les résultats de niveau de service. Les commentaires sur les causes initiales des problèmes sont associés aux métriques.

Il est également possible de générer des annotations d'événement qui sont des commentaires associés aux événements issus des données brutes plutôt qu'à la métrique. Vous pouvez afficher ces deux types de commentaires dans les données de rapport.

Deux méthodes dans l'objet "Tools" de la logique applicative tiennent compte de la création des causes initiales des problèmes et des enregistrements d'annotations d'événement :

- **Tools.AddRootCauseComment** (Text, Timestamp [resourceId])
- Enregistre un commentaire sur les causes initiales. Ces informations peuvent être utiles ultérieurement dans des rapports générés. Le commentaire sur les causes initiales enregistré décrit une situation spécifique au cours de l'exécution de la formule de la logique applicative à un moment donné. Le paramètre Information spécifie que le commentaire doit être écrit. La méthode reçoit un horodatage qui doit être enregistré avec le commentaire. Elle accepte également un paramètre ResourceId qui spécifie une ressource associée au contexte de méthode. (Ce paramètre est facultatif et peut être ignoré.)

- **Tools.AddEventAnnotation** (EventId, Text)
- Vous pouvez utiliser ces méthodes n'importe où dans la logique applicative, et le contexte d'application doit être pris en compte. Ajouter un commentaire sur les causes initiales peut être plus pertinent au terme d'une période de calcul, lorsque le motif pour un niveau de service inférieur aux attentes durant cette période est connu. Supposons que, par exemple, pendant la période d'un mois il y a eu quatre interruptions, toutes causées par un seul dispositif. Vous pouvez alors rassembler le commentaire sur les causes initiales à partir des informations relatives aux interruptions, de sorte à ce que lorsque les rapports sont consultés pendant cette période, les causes de la violation d'un niveau de service pendant cette période soient apparentes. La commande AddRootCauseComment est mieux adaptée à la routine du gestionnaire d'événements OnPeriodEnd, ou une autre fonction similaire qui est exécutée vers la fin de la période en cours de calcul.
- Cependant, ajouter une annotation d'événement convient mieux au traitement d'événement de données brutes réel et son utilisation est préférable à OnXXXEvent (gestionnaire d'événements personnalisé spécifié dans l'instruction d'enregistrement). Dans ce gestionnaire d'événements, tous les champs spécifiques à l'événement réel sont disponibles via l'objet eventDetails.
- Un exemple possible dans la logique applicative pourrait être comme suit :

```
Sub OnPeriodEnd(TIME)
        pctAvailable = (TotalTime-OutageTime) / TotalTime
        Tools.AddRootCauseComment "Violations caused by the
   following items: " & ViolationCollection, Time
End sub
…
…
Sub OnIncidentEvent(eventDetails)
    OutageTime = OutageTime + eventDetails("TimeToResolve")

    If eventDetails('TimeToResolve') > 6 Then
        ViolationCollection = ViolationCollection &
    eventDetails("HardwareId")
        Tools.AddEventAnnotation eventDetails.EventId, 'Incident ' _
          eventDetails('IncidentId') & ' not resolved within target
    time 6 hours"
    End If
End sub
```

Séparation des exceptions dans les intervalles de temps

La logique applicative CA Business Service Insight ne reçoit pas d'événements d'exception. Elle reçoit un OnTimeslotExit lorsque une période d'exception démarre et OnTimeslotEnter lorsque une période d'exception se termine. La logique applicative ne peut par conséquent pas faire la distinction entre les temps d'exception et les temps en dehors de la période d'application. En outre, elle ne peut pas établir de distinction entre des types d'exception. Par conséquent, il n'est pas possible d'implémenter une logique différente pour le comportement de temps d'exception et pour le comportement "en dehors de la période d'application".

Une façon d'implémenter des exceptions spéciales (c'est-à-dire, une exception qui ne se comporte pas comme une période "en dehors de la période d'application") est de définir des types d'événement dédiés, au lieu d'utiliser le mécanisme intégré de CA Business Service Insight pour le traitement des exceptions. Ces événements sont générés en les lisant depuis une source de données dédiée à l'aide d'un adaptateur.

Une feuille de calcul Excel (ou une autre source de données) peut stocker ces exceptions, et un adaptateur peut alors charger les données et générer une réponse : Evénements Exception Enter et Exception Exit. Les exceptions peuvent sinon être ajoutées à l'aide des corrections. En plus de la correction, une ressource fictive doit être définie et associée à ces événements à des fins d'enregistrement. Cette ressource n'a d'autre fonction que de servir d'espace réservé, tel qu'exigé par la commande.

Afin de pouvoir traiter les temps d'exception signalés par ces événements dédiés, la formule de la logique applicative doit s'enregistrer avec ces événements d'exception en plus de l'enregistrement normalement obligatoire des événements de données brutes à utiliser dans le calcul.

Il est recommandé que l'expert en logique applicative inclue un champ pour le type d'exception dans le type d'événement, en tenant compte du traitement des différents types d'exceptions particulières.

Cette approche présente les caractéristiques suivantes :

- Elle sépare deux jeux d'exception, l'exception standard et l'exception spéciale.
- Les exceptions spéciales n'ont pas d'interface utilisateur pour la maintenance.

- Les rapports basés sur des exceptions générées comme événements par un adaptateur ne commentent pas leur existence. Dans les cas où le mécanisme de correction a été utilisé, il y a un commentaire. L'utilisation de la méthode de correction est recommandée pour maintenir l'intégrité des résultats produits par le système.
- Aucune spécification "Avec/Sans exceptions" ne considère ces exceptions.
- Lorsque le mécanisme de correction est utilisé, la correction Avec/Sans s'applique.

Une fois la correction implémentée, il est recommandé que l'expert en logique applicative applique la logique à l'ensemble des métriques du système.

Une autre méthode permet d'appliquer une exception sur une ressource unique si nécessaire. Cette méthode implique d'utiliser le statut En vigueur des ressources. Définir le statut d'une ressource sur Pas en vigueur signifie que pendant cette période le moteur de calcul va ignorer tous les événements de données brutes qui sont envoyés pour cette ressource. En définissant une période de temps où la ressource n'est pas en vigueur en créant de nouvelles versions de la ressource, une au démarrage de la période d'exception, et une autre à la fin de la période d'exception.

Toutefois, si la ressource fait partie d'une métrique groupée et la ressource est à la fois en vigueur et pas en vigueur dans la même période de calcul, seule la dernière période où la ressource a été en vigueur sera prise en considération dans le résultat comme indiqué ci-dessus. Dans ce cas il est recommandé d'utiliser la fonctionnalité d'attributs personnalisés. Vous pouvez gérer un attribut supplémentaire pour la ressource indiquant que le statut de la ressource peut être généré et la formule de logique métier va interroger le statut de la ressource dans chaque emplacement pertinent du script.

Remarques concernant la mémoire et les performances

Ce qui suit est un ensemble de situations à prendre en compte lors de la conception des solutions de la logique applicative. Les situations décrites sont des cas où la performance du moteur de calcul peut avoir un impact négatif :

- **Paramètres**

  Paramètres : Si une valeur d'un paramètre est nécessaire dans le code, la création d'une variable globale pour affecter la valeur du paramètre est recommandée. De plus, quand la valeur du paramètre est nécessaire, utilisez à la place la variable globale. Cela évite une situation où le moteur crée le mappage des paramètres pour chaque appel de paramètre.

- **Utilisation des plans dans les métriques groupées**

  Les larges objets de mappage globaux de la logique applicative pour des métriques groupées devraient uniquement être utilisés avec une extrême précaution. Pendant le calcul par le moteur d'une métrique groupée, celui-ci charge les variables globales séparément dans le cluster à partir des états précédents pour chaque élément.

- **Enregistrement de la logique applicative**

  Il est recommandé de filtrer les événements de données brutes uniquement en fonction des méthodes d'enregistrement. L'ajout d'un filtre interne à l'aide d'une instruction "if" dans le code augmente le temps de traitement. En outre, des frais généraux supplémentaires sont utilisés par le moteur pour l'extraction et le traitement des enregistrements de données brutes qui ne sont pas nécessaires.

- **Eviter l'utilisation de Dispatcher.RegisterByEventType**

  Améliore les performances. L'utilisation de cette méthode d'enregistrement signifie que vous enregistrez toutes les ressources dans le système et pas uniquement les ressources qui contiennent des événements de ce type spécifique. Ainsi, tous les changements dans la ressource affectent les calculs de métrique. Un autre inconvénient lié à l'utilisation de cette méthode d'enregistrement apparaît au cours de l'exécution de la métrique lorsqu'elle accède aux données brutes. Elle doit filtrer dans les données brutes uniquement les événements contenant le type d'événement spécifique et ignorer les autres événements.

- **Dispatcher.Register**

  Lorsque vous utilisez Dispatcher.Register, vérifiez *toujours* que vous spécifiez le 3e paramètre. L'enregistrement sans le 3e paramètre revient exactement à enregistrer par type d'événement (Dispatcher.RegisterByEventType). En d'autres mots, vérifiez que vous utilisez au moins un autre paramètre en plus des deux premiers.

- **Granularité de calcul**

  Il est important d'activer uniquement les agents nécessaires à des fins de calcul et d'exploration vers le bas. Le calcul de toutes les unités de temps de l'agent sollicite fortement les ressources du processeur.

Activation de contrats (gestionnaire de contrats)

L'activation d'un contrat s'effectue en le validant. (Pour d'autres détails, consultez l'aide en ligne.)

L'activation du contrat déclenche le moteur pour initier les calculs de la métrique du contrat et pour commencer à produire des résultats pour le contrat.

Avant d'activer la vérification du contrat, vérifiez si toutes les conditions suivantes ont été remplies :

- Toutes les métriques doivent avoir leur logique applicative définie (dans la métrique ou comme module lié) qui a été testée et ne contient pas d'erreurs.
- Toutes les métriques ont un seuil de tableau de bord défini. (Pour d'autres détails,consultez le Manuel de l'utilisateur.) Il est important que les seuils soient déjà définis pour que le tableau de bord puisse évaluer les limites de la métrique pendant le processus du calcul.
- Les dates d'entrée en vigueur correspondent à la période correcte et des enregistrements de données brutes sont disponibles. Vérifiez également que les résultats sont conformes en utilisant la portée de la logique applicative.

Une fois que le contrat a été validé, vérifiez si l'activation a commencé correctement et que les calculs progressent comme prévu.

Procédez comme suit :

1. Vérifiez que tous les composants de service CA Business Service Insight ont été lancés, notamment le moteur de calcul, qui inclut aussi bien PSLWriter que PenaltyWriter. Il est recommandé que tous les composants de service s'exécutent quand le calcul est nécessaire.
2. Diagnostiquez le contrat : La fonction de diagnostic du contrat affiche les résultats d'une série de tests accomplis sur toutes les métriques de contrat (et les formules de pénalité le cas échéant). La sévérité du résultat de test est fournie en même temps qu'une procédure de résolution recommandée. Il est recommandé d'accomplir un diagnostic lors de l'activation d'un contrat ainsi qu'au terme du calcul pour ce contrat.
3. Générez le rapport au format libre 'Statut du calcul'. Ce rapport est intégré dans l'installation initiale de CA Business Service Insight et se trouve dans le dossier de groupement des rapports d'administration. Il fournit des informations sur la progression du calcul et peut être utilisé à ce stade pour vérifier si le moteur PSL progresse et si le calcul est terminé. Vérifiez ce rapport pour évaluer si des problèmes peuvent affecter les calculs.

Le rapport contient les champs de colonne suivants :

Champ	Description
Contrat	Indique le nom du contrat. La liste contient les noms des contrats qui sont en vigueur et qui ne sont pas en vigueur.
Métrique	Indique le nom de la métrique dans le contrat. La liste contient toutes les métriques contenues dans chaque contrat.
Période de suivi	Indique la période de calcul de la métrique. La liste affiche une entrée pour chaque unité de temps de calcul de la métrique en fonction des agents qui sont actifs et sur la base de la définition de granularité de la métrique. Si la période de calcul est la période de suivi, cela est spécifié.
Mise à jour jusqu'à	Indique le moment de la dernière mise à jour. Cela montre que le résultat pour la métrique spécifique est disponible jusqu'à cette date. Par exemple, si ce champ affiche 01/01/2006, cela indique que tous les résultats pour cette métrique dans cette unité de temps sont mis à jour jusqu'à cette date et que les rapports sont disponibles jusqu'à cette date.

Début du dernier cycle	Indique l'heure du cycle lorsque le calcul de mise à jour du résultat de la métrique a commencé.
Requiert un nouveau calcul depuis	Si le moteur marque une métrique donnée pour un nouveau calcul et que celle-ci n'a pas encore été mise à jour, la date résultante s'affiche ici pour signaler le moment depuis lequel les résultats ne sont plus pertinents et nécessitent une mise à jour. Ce peut être le cas pour tous les nouveaux calculs.
Dernière mise à jour à	Heure à laquelle le moteur a mis à jour l'enregistrement en fonction du dernier résultat.
Traité par le moteur #	Indique le numéro du moteur affecté pour gérer le traitement du calcul de la métrique spécifique.

Ce rapport peut également fournir les informations suivantes d'après les données brutes disponibles :

- Le temps qu'il a fallu au moteur pour calculer une métrique unique. Il est possible de voir combien de temps il a fallu au moteur pour mettre à jour chaque métrique, en triant toutes les métriques qui ont été calculées dans un cycle unique en fonction de leur heure de mise à jour. Tous les enregistrements présentant la même valeur "Début du dernier cycle" sont calculés pendant le même cycle et l'heure de mise à jour est l'heure de la "Dernière mise à jour à". Il est possible d'évaluer le temps qu'il a fallu au moteur pour calculer la métrique entière avec les unités de temps de son agent sous-jacent ainsi que chacune des unités temps.
- Le temps qu'il a fallu au moteur pour calculer un contrat complet. Ce calcul est effectué en examinant l'heure de mise à jour de la première métrique du contrat et l'heure de la dernière mise à jour de la dernière métrique de ce contrat, pour ensuite calculer le délai entre les deux.

Nouveau calcul complet des métriques du contrat

Le processus du nouveau calcul dans le contexte actuel est l'initialisation d'un nouveau calcul de système complet, réalisé soit par l'expert en sources de données, soit par l'expert en logique applicative. Il ne s'agit pas du nouveau calcul de moteur accompli pendant un processus de calcul standard. Ce type d'action s'accomplit habituellement pendant ou après le processus de configuration du contrat, lors duquel divers mauvais fonctionnements peuvent avoir été détectés.

Il est recommandé de commencer un nouveau calcul complet uniquement une fois que le système est stable (c'est-à-dire, pas pendant la création du système), avant de passer en direct sur le site.

Le nouveau calcul est actuellement fait en exécutant un script SQL sur la base de données. Ce script nettoie la table PSL et toutes les tables d'accompagnement connexes qui font partie du processus de calcul. Ce script doit être approuvé et validé par l'équipe d'assistance de CA avant d'être exécuté, les modifications de structure occasionnelles apportées au système ayant pu aboutir à des changements au niveau du schéma de la base de données et/ou d'objets dépendants.

Remarque : Avant d'exécuter le script, tous les composants de service doivent être arrêtés et, après l'exécution, ils peuvent être redémarrés afin de relancer les calculs.

Les situations suivantes peuvent indiquer la nécessité d'un nouveau calcul complet :

- Problèmes relatifs aux définitions dans le contrat : si à cette étape, il est découvert qu'une cible de métrique a incorrectement été définie ou qu'un seuil de métrique est erroné.
- Erreurs dans le contrat.
- Une condition pour réévaluer les seuils de tableau de bord ou pour régénérer des alertes SLA.

Remarque : Contactez le centre d'assistance de CA pour vous entretenir de cette option, le cas échéant, et pour obtenir des copies des scripts pertinents pour la version d'CA Business Service Insight installée.

Création de livrables (gestionnaire de contrats)

A cette étape, toutes les sorties du système sont configurées pour répondre aux exigences, y compris la création et le formatage de tous les rapports.

La configuration de livrables inclut ce qui suit :

- Définition des paramètres de sécurité (autorisations et groupes d'utilisateurs).
- Création de rapports enregistrés.
- Création de livrets.
- Création de modèles d'exportation de contrat.
- Création de vues de navigateur des prestations de services.
- Configuration de plans de tableau de bord.
- Création de profils d'alerte de niveau de service.

Définition des paramètres de sécurité (administrateur)

Lors de la définition des paramètres de sécurité, les utilisateurs CA et leurs autorisations connexes doivent être configurés. Ces paramètres incluent la définition des informations devant être accessibles à un utilisateur (les entités au sein du système que l'utilisateur peut afficher ou modifier). Ces autorisations peuvent être définies à divers niveaux, selon des groupes d'utilisateurs, des rôles ou même par utilisateur. Les informations accessibles sont définies par rapport à des contractants et peuvent être définies directement selon l'utilisateur ou héritées du groupe d'utilisateurs auquel l'utilisateur appartient.

C'est à cette étape que les rôles principaux sont configurés et que les groupes d'utilisateurs leur sont associés, pour que lors de l'ajout d'un nouvel utilisateur, celui-ci soit rattaché uniquement à un groupe afin d'hériter des paramètres pertinents.

Les actions autorisées sont configurées dans les rôles et sont fournies à l'utilisateur par association directe de l'utilisateur à un rôle, ou par le groupe d'utilisateurs auquel l'utilisateur appartient. Les actions associées au tableau de bord autorisées sont également définies dans le rôle associé.

Il est recommandé que l'administrateur détermine les groupes d'utilisateurs et les rôles qui doivent être définis, ainsi que leur autorisation requise afin de pouvoir mettre en place une structure de prise en charge facile pour l'ajout d'utilisateurs.

Création de rapports

Utilisez le processus suivant pour créer un rapport :

1. Créez tous les dossiers requis pour le rapport-tous les dossiers doivent être créés d'avance afin d'être disponibles lors de l'enregistrement de chaque nouveau rapport enregistré. Habituellement, chaque contrat dispose de son propre dossier, y compris un dossier exécutif pour les rapports de haut niveau.
2. Définissez les critères de filtrage du rapport à l'aide de l'assistant de création de rapports-chaque création de rapport enregistré démarre en générant le rapport via l'assistant de création de rapports. Dans l'assistant de création de rapports, les critères de filtrage requis sont sélectionnés et le rapport est généré. Les administrateurs informatiques peuvent définir des paramètres de rapport en désignant des champs définis par l'utilisateur que la personne qui utilise/consulte le rapport doit remplir, permettant ainsi la génération de résultats de rapport propres à l'intérêt de cet utilisateur.

   Lorsque vous générez un rapport afin de le définir comme rapport enregistré, il est très important de définir les critères de filtrage de manière flexible. Cela évite tout travail inutile au fur et à mesure que le système prend de l'ampleur et évolue. Chaque rapport doit contenir des informations actuelles et mises à jour. Par exemple, si un rapport indique actuellement trois composants de service, à l'avenir, lors de l'ajout de nouveaux composants de service, il est important que ce service s'ajoute automatiquement au rapport et n'exige aucune nouvelle configuration de rapport. Ou encore, en cas d'établissement de rapport par mois et si le rapport indique actuellement trois mois, le mois suivant, les quatre mois, y compris le dernier mois, doivent être indiqués. Dans de nombreux cas, les rapports doivent toujours indiquer la valeur des 6 derniers mois de données. Ces types de rapport à fenêtre glissante sont extrêmement utiles contrairement aux rapports à durée fixe, puisqu'ils n'exigent aucune attention supplémentaire une fois créés.

Les quelques astuces suivantes sont utiles pour définir les critères de filtrage des rapports enregistrés :

Onglet Critères

- Le recours à l'option Données commerciales uniquement limite les données affichées uniquement à celles qui sont associées à la période de suivi de la métrique.
  - Privilégiez toujours l'option *Regrouper par* ou *Rapport par* plutôt que de sélectionner les métriques spécifiques sur lesquelles établir un rapport.

- Définissez une plage horaire dynamique. Si cela est défini comme plage horaire fixe, le rapport indiquera toujours les mêmes résultats. (Autrement dit, 3 derniers mois est une plage dynamique.)

Onglet Divers

- Vérifiez s'il faut présenter les "périodes incomplètes" dans le rapport. S'il le faut, sélectionnez alors cette option à partir de la liste. Habituellement, lors de la configuration des rapports, la période incomplète est exclue, car comme il ne s'agit pas d'un résultat final, ce n'est pas un résultat professionnel.
- Concevez la mise en page du rapport-une fois le rapport généré, il est possible de concevoir sa mise en page à l'aide de l'outil de conception de la rage Rapport. Le modèle peut avoir des critères propres au destinataire prévu pour le rapport. Il est recommandé de créer plusieurs mises en page de conception, une pour chaque scénario possible, et d'appliquer ces conceptions au reste des rapports à générer. Par conséquent, lors de la définition des critères du rapport, dans l'onglet *Afficher,* choisissez *Utiliser la conception de*.

  Remarque : Pour obtenir des astuces sur l'utilisation de l'outil d'édition pour concevoir la mise en page des rapports, consultez la section suivante.

Onglet Général

- Enregistre le rapport. Une fois le rapport généré et conçu, il peut être enregistré dans le dossier pertinent.
- Pendant le processus d'enregistrement du rapport, celui-ci est associé aux utilisateurs qui y ont accès. Il est par conséquent important que le groupe d'utilisateurs soit déjà défini pour qu'il soit possible d'associer les rapports aux utilisateurs. Les utilisateurs dotés d'autorisations adéquates peuvent être associés au rapport ultérieurement à l'aide des options de dossier.
- Associez les rapports reliés pour que la navigation entre des rapports similaires ou présentant un lien commun, soit simplifiée pour les utilisateurs de ces rapports.
- La page Dossier de rapports vous permet de créer des rapports, des groupes de rapport, des rapports composés ou au format libre, des livrets, des raccourcis et des dossiers, mais aussi de chercher des rapports.

  A partir du menu Rapports, cliquez sur Dossier de rapports. La page Dossier de rapports s'ouvre et affiche une liste des rapports enregistrés.

Rapport de déviation

La valeur de déviation est calculée automatiquement par le moteur CA Business Service Insight pour les métriques qui ont une cible. La valeur représente la différence entre le niveau de service réel et la cible. La formule de calcul de la déviation, calculée automatiquement par CA, change en fonction de la définition du niveau de service : si la cible de niveau de service est une valeur maximale (lorsque le niveau de service réel est "Pas plus de") ou une valeur minimale (lorsque le niveau de service réel est "Pas moins de"). Voir ci-dessous un exemple de changement de la formule :

Instruction Cible	Seuil de niveau de service	Formule de déviation
Le service doit être disponible au moins 99 % du temps programmé.	La cible est le seuil minimal.	$Service_{Dev} = (-1)x \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$
La durée moyenne de réparation (MTTR) ne doit pas dépasser 4 heures par mois.	La cible est le seuil maximal.	$Service_{Dev} = \frac{Service_{SL} - Service_{SLT}}{Service_{SLT}}$

Où $Service_{Dev}$ = Déviation du service

Où $Service_{SLT}$ = Performances attendues du service

Où $Service_{SL}$ = Performances réelles du service

L'exemple suivant illustre un calcul de déviation minimale :

Le service doit être disponible au moins 99 % de la période d'application prévue. Le niveau de service réel est de 98 % pendant la période d'application prévue.

$$Service_{Dev} = (-1)x \frac{98\% - 99\%}{99\%} = 1.0101\%$$

Les rapports de déviation permettent de disposer de vues de haut niveau des garanties d'une nature étrangère (type différent de calcul) et pour les rassembler dans une barre unique avec des points communs.

Si, par exemple, la matrice suivante figure dans le contrat :

Service-Centre d'assistance	**Domaine de service-Gestion de ticket Priorité 1**	**Domaine de service-Gestion de ticket Priorité 2**	**Domaine de service-Gestion de ticket Priorité 3**
	Temps de résolution moyen P1	Temps de résolution moyen P2	Temps de résolution moyen P3
	% P1 de tickets résolus en T1	% P2 de tickets résolus en T1	% P3 de tickets résolus en T1
	% P1 de tickets répondus en T1	% P2 de tickets répondus en T1	% P3 de tickets répondus en T1
	Temps de réponse moyen P1	Temps de réponse moyen P2	Temps de réponse moyen P3

Les résultats de la génération d'un rapport de domaine de service filtré par service de centre d'assistance se présentent comme dans le graphique suivant.

Le rapport susmentionné permet au gestionnaire de niveau de service de voir les performances du centre d'assistance en fonction de diverses priorités, sans tenir en compte du contrat ou du type d'obligation.

Toutes les métriques du centre d'assistance sont rassemblées dans une barre unique en fonction de leur priorité.

Par exemple, la barre de Priorité 1 affiche les trois métriques définies dans la métrique et rassemble leur déviation en une valeur unique. L'assistant de création de rapports permet de choisir la méthode agrégative, à savoir une moyenne, un minimum ou un maximum.

Un tel rapport permet au gestionnaire de conclure, par exemple, qu'il a besoin d'investir plus de ressources dans les tickets désignés Priorité 1 ou de changer les contrats qui s'y rapportent.

Cet exemple indique que la modélisation fournit aussi bien le rapport sur une obligation unique, qui indique si un contrat a été honoré ou rompu, qu'un rapport de gestion plus général qui permet au gestionnaire de niveau de service de gérer ses ressources plus efficacement et d'améliorer ainsi ses composants de service.

Rapports au format libre

Les rapports au format libre permettent aux utilisateurs de générer des rapports d'après des requêtes SQL de la base de données CA Business Service Insight ou d'une autre source de données externe, qui peut être accédée via une connexion du serveur CA Business Service Insight. Cela inclut également tout autre type de source de données à laquelle il est possible d'accéder via ODBC, comme Excel, Access, Lotus Notes, les fichiers texte, etc. Les rapports au format libre sont souvent utilisés pour configurer des rapports statistiques d'après des données créées par les commandes de logique applicative Tools.SaveRecord et Tools.SaveFields.

Les rapports au format libre se connectent via une chaîne de connexion à une base de données sélectionnée et exécutent une requête SQL sur la base de données en utilisant une chaîne de requêtes. Il est possible d'ajouter des paramètres aux deux chaînes afin de créer des rapports dynamiques qui permettent à l'utilisateur de saisir ou de sélectionner des valeurs spécifiques à inclure dans la requête, comme le nom et le mot de passe de l'utilisateur pour se connecter à la base de données.

Les rapports au format libre sont affichés dans les onglets Graphique, Donnée et Filtrer comme le sont les rapports générés à l'aide de l'assistant de création de rapports.

Remarque : Les rapports au format libre peuvent inclure des graphiques, uniquement si toutes les colonnes, en dehors de la première colonne, sont numériques. Les données figurant dans la première colonne permettent de nommer l'axe X. Le nom des colonnes est utilisé pour les autres titres.

Du fait que les rapports au format libre utilisent un accès direct à une base de données et à une requête SQL ouverte, la maintenance est problématique. La plus grande prudence s'impose pour ne pas affecter les données sous-jacentes qui servent de source pour les rapports au format libre. Lorsque les rapports sont générés à partir d'une source de données externe, il est recommandé de configurer un processus de notification pour assurer que ces sources de données ne sont pas sujettes à modifications sans d'abord avoir consulté le gestionnaire de contrats responsable des rapports de données au format libre.

Informations générales à garder à l'esprit lors de la création de rapports au format libre.

- En raison de problèmes de formatage international des dates, il est souvent utile de préciser le format exact pour la date en créant une formule personnalisée dans la logique applicative. Cela permet de convertir la date en une chaîne du même format chaque fois et évite les problèmes de formatage de dates entre les Etats-Unis et l'Union européenne. Cette méthode doit être utilisée pour les dates inscrites dans la table T_SLALOM_OUTPUTS via les commandes Tools.SaveFields ou Tools.SaveRecord. La formule modèle est fournie ci-dessous :

Fonction FormatDate(DateField)

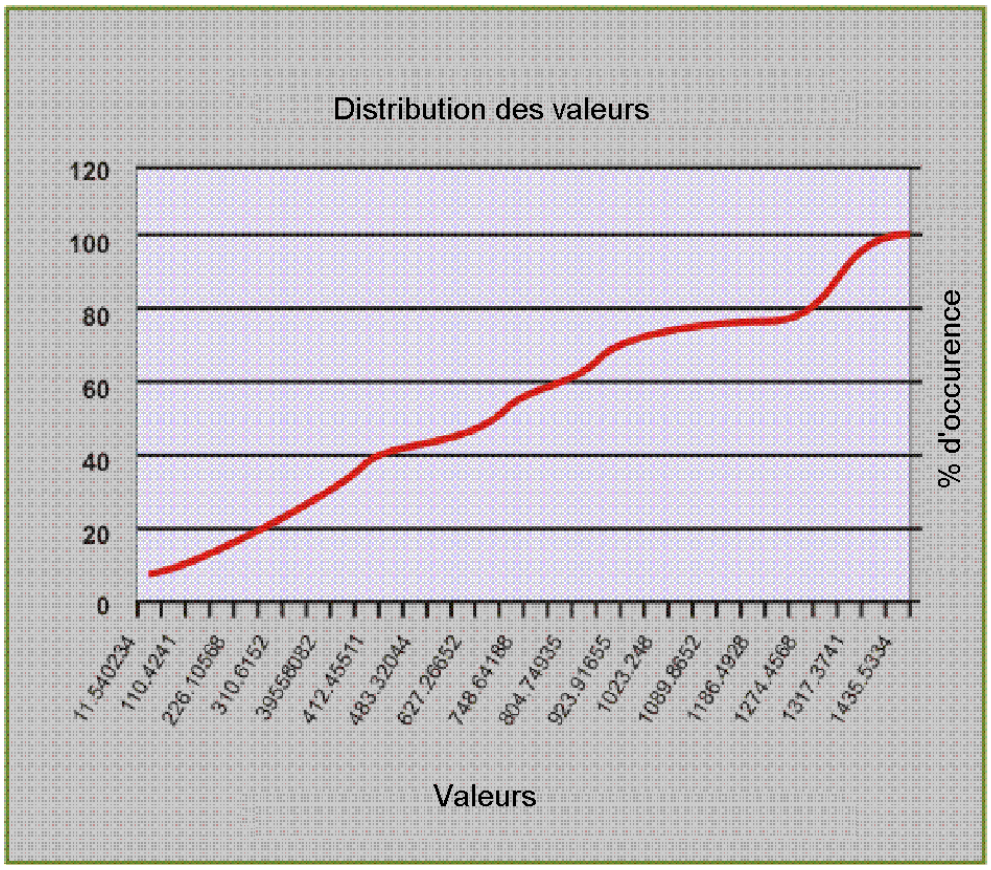
```
    If DateField = "" Then
      FormatDate = ""
    Else
      Dim PeriodYear, PeriodMonth, PeriodDay, PeriodHour,
  PeriodMinute, PeriodSecond
      PeriodYear    = DatePart("yyyy",DateField)
      PeriodMonth   = DatePart("m",DateField)
      PeriodDay     = DatePart("m",DateField)
      PeriodHour    = DatePart("m",DateField)
      PeriodMinute  = DatePart("m",DateField)
      Periodsecond  = DatePart("s",DateField)
      FormatDate  = PeriodDay&"/"&PeriodMonth&"/"&PeriodYear& _
 " "&PeriodHour&":"&PeriodMinute&":"&Periodsecond
    End If
End Function
```

- Lors de l'emploi du paramètre heure des gestionnaires d'événements de la logique applicative ou de tout horodatage de l'objet eventDetails, tenez toujours compte de l'impact du changement horaire. Soyez conscient que, lors du report de dates dans la table, l'heure peut être fournie au format UTC plutôt qu'en heure locale. Il vous faudra peut-être utiliser la fonction Tools.GetLocaleTime() pour y remédier.
- Vous pouvez toujours utiliser l'utilitaire de conception de rapport (lorsqu'un rapport au format libre produit un graphique) pour personnaliser l'aspect.
- Les options d'exportation au format PDF pour le rapport au format libre sont personnalisables via la section Paramètres de rapport de la fenêtre de configuration du format libre. Vous pouvez, par exemple, personnaliser la mise en page (portrait ou paysage).
- Vous pouvez incorporer du code HTML dans les rapports pour exécuter différentes fonctionnalités, comme ajouter des liens hypertexte ou changer les couleurs de colonne et de ligne, les polices ou les paramètres d'affichage.
- Vous pouvez créer des vues de base de données (fonctionnalité Oracle) dans la table T_SLALOM_OUTPUTS, afin de simplifier le langage SQL requis pour les rapports.

- Lors de la spécification de paramètres pour les rapports, vous pouvez les définir de diverses façons. Par exemple : texte libre, numérique (valider), mot de passe (caractères masqués - utile pour les mots de passe des connexions de base de données), date (en utilisant le sélecteur de dates fourni) et listes (créées en spécifiant une instruction SQL pour remplir la liste).
- Les valeurs de paramètre spécifiées dans la section SQL du rapport au format libre doivent être remplacées par le nom de paramètre, précédé du symbole @. Par exemple : @PARAM1. L'utilisation de guillemets pour encadrer cette valeur de paramètre peut être exigée s'il s'agit d'une chaîne.

Rapports au format libre - Histogrammes génériques

La requête suivante peut être utilisée dans un rapport au format libre afin de présenter la distribution de valeurs dans un tableau par pourcentage, comme illustré dans le graphique suivant :



Vous pouvez voir dans le graphique susmentionné la proportion (en pourcentage) des valeurs inférieures à 11,5 (0 %), à 804,74 (~50 %) et à 1435,53 (100 %).

Si le SLA spécifie des cibles comme "x % des valeurs doivent être inférieurs à y", les résultats de ce format libre contribuent à identifier les valeurs x et y qui assurent la conformité avec le SLA.

Les paramètres utilisés dans la requête sont les suivants :

- **@Query** : une instruction Select sous la forme suivante : select *valeur* **val** from *table*. L'alias val est obligatoire. Cette requête fournit les valeurs pour lesquelles la distribution est analysée.
- **@Buckets** : nombre de valeurs sur l'axe des x. Les valeurs sources sont arrondies à ces chiffres. Par exemple, si vous spécifiez @Buckets = 100, les valeurs des données sources sont divisées en 100 groupes et la requête calcule le nombre de valeurs affectées à chaque groupe. Plus @Buckets est élevé, plus le résultat est précis, mais le graphique n'est pas plaisant à regarder.
- **@Relation** : sens de l'accumulation. Valeurs possibles : "<=" (lorsque la cible n'est pas inférieure à) ou ">=" (dans les autres cas).

La requête peut être exécutée par rapport à la source de données ou T_SLALOM_OUTPUTS pour obtenir les meilleurs résultats.

La requête suivante produit le graphique ci-dessus :

```
select val,100*records/(select count(*) from (@Query))
from
(
  select x.bucket_val val,
    sum(y.records) records
  from
  (
    select round(val/bucket_size,0)*bucket_size bucket_val,
      count(*) records
    from
    (
      select (max(val)-min(val))/@Buckets bucket_size
      from
      (
        @Query
      )
    ) params,
    (
          @Query
       ) source
```

```
    group by round(val/bucket_size,0)*bucket_size
    order by round(val/bucket_size,0)*bucket_size
  ) x,
  (
    select round(val/bucket_size,0)*bucket_size bucket_val,
      count(*) records
    from
    (
      select (max(val)-min(val))/@Buckets bucket_size
      from
      (
        @Query
         )
    ) params,
    (
            @Query
       ) source
    group by round(val/bucket_size,0)*bucket_size
    order by round(val/bucket_size,0)*bucket_size
  ) y
  where y.bucket_val @Relation x.bucket_val
  group by x.bucket_val
  order by x.bucket_val
)
```

Voici une liste de paramètres échantillons (comme XML) pouvant être utilisés :

```
<custom>
     <connection>
          <params/>
     </connection>
     <query>
          <params>
               <param name="@Query" disp_name="Data Type" type="LIST">
                    <value>Réussite de l'activation du contexte PDP</value>
                    <list>
                         <item>
                              <value>select success_rate as val from
PDP_Context_Activation_Success.CSV</value>
                              <text>Réussite de l'activation du contexte
PDP</text>
                         </item>
                         <item>
                              <value>select throughput as val from [gprs
throughput volume by apn.csv]</value>
                              <text>Débit d'un APN unique</text>
                         </item>
                         <item>
                              <value>select throughput as val from [Generic GPRS
Throughput.CSV]</value>
```

```
                                        <text>Débit générique</text>
                                   </item>
                              </list>
                         </param>
                         <param name="@Buckets" disp_name="X Axis Values" type="LIST">
                              <value>100</value>
                              <list>
                                   <item>
                                        <value>25</value>
                                        <text>25</text>
                                   </item>
                                   <item>
                                        <value>50</value>
                                        <text>50</text>
                                   </item>
                                   <item>
                                        <value>100</value>
                                        <text>100</text>
                                   </item>
                                   <item>
                                        <value>250</value>
                                        <text>250</text>
                                   </item>
                                   <item>
                                        <value>500</value>
                                        <text>500</text>
                                   </item>
                                   <item>
                                        <value>1000</value>
                                        <text>1000</text>
                                   </item>
                              </list>
                         </param>
                         <param name="@Relation" disp_name="Violation of threshold means"
type="LIST">
                              <value>résultat insuffisant</value>
                              <list>
                                   <item>
                                        <value>&gt;=</value>
                                        <text>résultat insuffisant</text>
                                   </item>
                                   <item>
                                        <value>&lt;=</value>
                                        <text>résultat excessif</text>
                                   </item>
                              </list>
                         </param>
                    </params>
               </query>
```

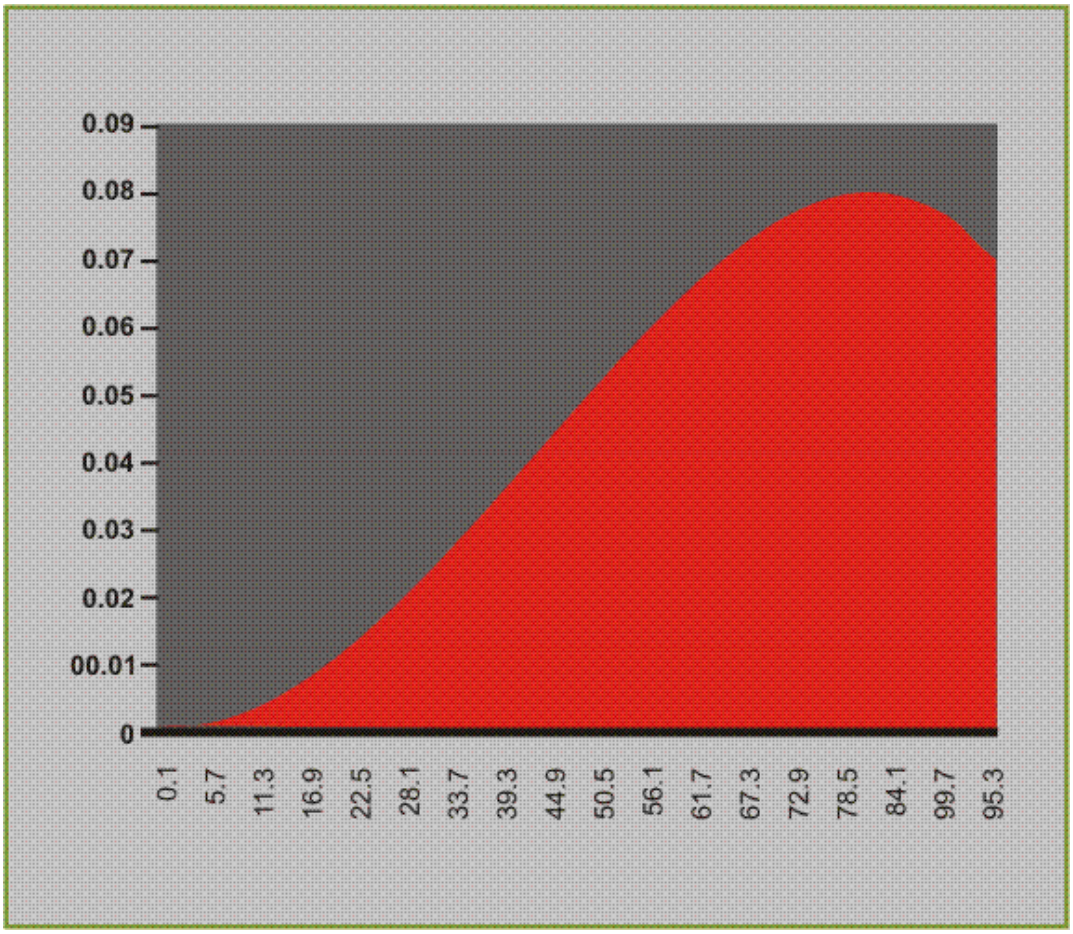
```
</custom>
```

Commentaires

- La requête a été optimisée pour Oracle et SQL Server. Pour d'autres sources de données ODBC, il peut être nécessaire d'ajouter "as" devant les alias de colonne et d'appliquer d'autres modifications.
- Lors de l'exportation de résultats dans Excel, il est recommandé que l'utilisateur génère un rapport XY (nuages de points) pour le représenter. Lorsque le graphique est en pointillé, il est également possible de voir la dispersion des valeurs.

Rapport au format libre - Distribution normale générique (type gaussien)

La requête détaillée ci-dessous peut être utilisée dans un rapport au format libre pour présenter la distribution normale de valeurs dans un tableau, comme illustré dans le graphique suivant :



Les paramètres utilisés dans la requête sont les suivants :

- @Query : une instruction Select sous la forme suivante :

```
select min (something) min_val,
          max (something)max_val,
          avg (something) avg_val,
          stddev (something) stddev_val
from table_name
```

  x_val est obligatoire. Cette requête fournit les valeurs pour lesquelles la distribution est analysée.

- @Buckets : nombre de valeurs sur l'axe des x. Les valeurs sources sont arrondies à ces chiffres. Par exemple, si vous spécifiez @Buckets = 100, les valeurs des données sources seront divisées en 100 groupes et la requête indiquera la valeur de distribution normale pour chaque groupe. Plus @Buckets est élevé, plus le résultat est précis, mais le calcul est plus lourd. 100 est un bon choix pour @Buckets.

La requête peut être exécutée par rapport à la source de données ou T_SLALOM_OUTPUTS pour obtenir les meilleurs résultats.

La requête suivante produira le graphique ci-dessus :

```
select(max_val-min _val)/@Buckets*serial,
     (1/stddev_val*sqrt (2*3,14159265359))*exp ( -1/(2*power (stddev_val,
2))*power (((max_val-min_val)/@Buckets*serial-avg_val), 2))
from     (
          @Query
     ),
     (
          select     rownum serial
          from  t_psl
          where  rownum < @Buckets
     )
order by serial
```

Ses paramètres XML correspondants :

```
<custom>
     <connection>
          <params/>
          </connection>
     <query>
          <params>
               <param name="@Query" disp_name="Data Type" type="LIST">
                    <value>Réussite de l'activation du contexte PDP</value>
                    <list>
                         <item>
<value>
```

```
select min(success_rate) min_val,max(success_rate) max_val,avg(success_rate)
avg_val,stddev(success_rate) stddev_val
       from PDP_Context_Activation_Success.CSV
</value>
                              <text>Réussite de l'activation du contexte
PDP</text>
                         </item>
                         <item>
                              <value>
select min(throughput) min_val,max(throughput) max_val,avg(throughput)
avg_val,stddev(throughput) stddev_val
       from [gprs throughput volume by apn.csv]
</value>
                              <text>Débit en Ko</text>
                         </item>
                         </list>
               </param>
               <param name="@Buckets" disp_name="X Axis Values" type="LIST">
                    <value>100</value>
                    <list>
                         <item>
                              <value>25</value>
                              <text>25</text>
                         </item>
                         <item>
                              <value>50</value>
                              <text>50</text>
                         </item>
                         <item>
                              <value>100</value>
                              <text>100</text>
                         </item>
                         <item>
                              <value>250</value>
                              <text>250</text>
                         </item>
                         <item>
                              <value>500</value>
                              <text>500</text>
                         </item>
                         <item>
                              <value>1000</value>
                              <text>1000</text>
                         </item>
                    </list>
               </param>
          </params>
     </query>
</custom>
```

Commentaires

- La requête a été optimisée pour Oracle et SQL Server. Pour d'autres sources de données ODBC, il peut être nécessaire d'ajouter "as" devant les alias de colonne et d'autres modifications.
- Lors de l'exportation de résultats dans Excel, il est recommandé que l'utilisateur génère un rapport XY (nuages de points) ou un rapport de zone pour le représenter.

Configuration des pages de tableau de bord

La section suivante fournit des recommandations sur le contenu à configurer pour les utilisateurs de CA Business Service Insight. La recommandation est au niveau élevé et doit tenir compte des exigences spécifiques du client. Les pages sont décrites dans les paragraphes suivants dans le contexte d'un utilisateur spécifique qui est décrit via son rôle dans l'organisation.

Pages de gestionnaire exécutif

Un gestionnaire exécutif est probablement intéressé par une vue de haut niveau de tous les départements, pays, comptes, etc. N'étant généralement pas opérationnel, il a besoin de vues qui lui fournissent des informations pour prendre des décisions stratégiques. Par conséquent, il peut être plus pertinent d'afficher le statut contractuel plutôt que le statut actuel dans les plans et la vue de tableau de bord agrégatif du gestionnaire exécutif.

Par exemple, les vues de tableau de bord suivantes peuvent être incluses sur la page Présentation :

Vue de tableau de bord	Description
Comptes critiques	Inclut tous les contractants qui sont marqués comme sensibles. Le gestionnaire exécutif sélectionne les contrats ou les contractants qu'il considère comme sensibles.
Performances globales	Inclut des widgets personnalisés de qualité globale, des widgets de vue agrégative qui incluent tous les indicateurs-clés de qualité des comptes.
Départements	Utilise une image d'arrière-plan avec le graphique organisationnel et place les widgets sur les départements pertinents. Généralement, le groupe de composants de service est utile ici en fonction de ce que représente un département dans l'organisation.

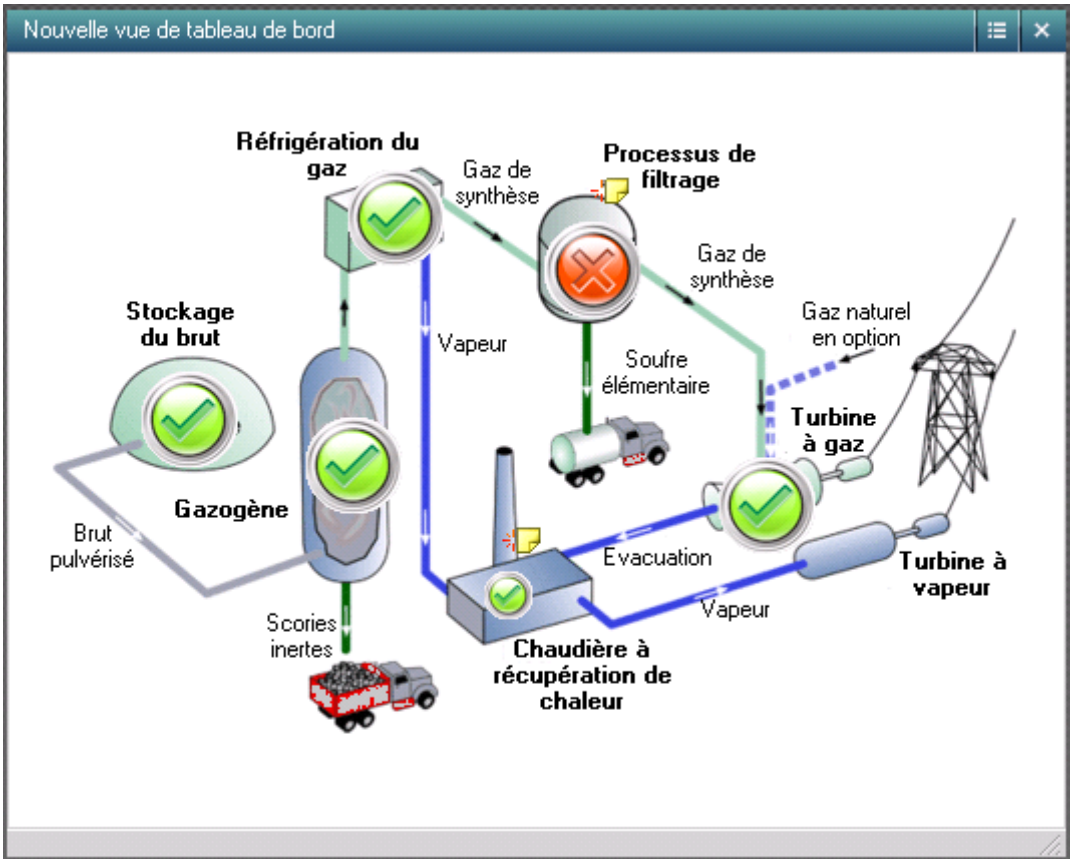
Données géographiques	Utilise une image d'arrière-plan avec une carte géographique et localise des widgets de groupes de contractants aux emplacements pertinents.
Performance financière	Inclut des widgets qui comprennent des informations agrégatives sur les métriques financières.
URL	Inclut des pages du portail d'entreprise, notamment des prospects dans l'équipe de vente.

Rapports recommandés pour la page :

Rapport	**Description**
Rapport de déviation	Reprend les dix pires contrats pour une certaine période, fournit des informations sur les zones qui sont les moins performantes sur le plan de la prestation de services.
Rapport financier pour le mois en cours	Résume l'état financier cumulé au fil du temps.

Page de processus :

cette page doit contenir une vue de tableau de bord qui présente un diagramme de processus doté de widgets représentant chaque chaîne dans le processus comme dans l'exemple ci-dessous :



Pages du gestionnaire de contrats

Les pages configurées pour un gestionnaire de contrats indiquent la qualité du service qui est assuré pour chacun des comptes qu'il gère. Vue de tableau de bord des contrats pertinents afin de mettre le gestionnaire au courant du niveau de service actuel fourni sur les obligations des contrats qui sont sous sa responsabilité. De plus, ce genre de page affiche les rapports disponibles pour chacun des composants de service qui sont inclus dans le contrat.

La page Présentation inclut les vues de tableau de bord suivantes :

Vue de tableau de bord	Description
Mes comptes	Widgets de tous les contrats ou contractants qui sont sous la responsabilité du gestionnaire de contrats.
Mes services	Composants de service à travers les comptes gérés par le gestionnaire de contrats.
Performances financières	Widgets qui incluent des informations agrégatives sur les métriques financières pour les comptes gérés.
URL	Portail comme le système CRM

Créez une page de compte pour chacun des comptes gérés qui fournissent une vue sur les obligations de compte spécifiques. Il est recommandé de combiner des vues de calcul du statut actuel au statut contractuel afin de donner au responsable de compte la capacité d'être proactif et d'agir rapidement. Par exemple :

Page de compte	Description
Obligations de compte	Inclut des métriques incluses dans le contrat.
Performances financières	Widgets qui incluent des informations agrégatives sur les métriques financières pour les comptes gérés.

Pages du gestionnaire de services

Pages à configurer pour un gestionnaire en charge d'un service ou d'un domaine particulier, qui affichent une vue détaillée des objectifs de service au travers des divers contrats et des états de ses objectifs. En outre, une telle page inclut des rapports qui mettent en surbrillance les paramètres de service critiques en cours de mesure.

Les vues de tableau de bord incluses dans les pages du gestionnaire de service sont similaires aux pages du responsable de compte, où seules les informations sont affichées et rassemblées dans un niveau de service plutôt qu'au niveau du contrat ou du contractant.

Pages d'accueil

Une page du tableau de bord peut être affectée à une page d'accueil, afin de disposer d'une passerelle personnalisable pour interagir avec le système, offrant un accès rapide aux informations et aux actions.

Ajout de profils d'alerte de niveau de service

Le profil d'alerte permet de définir les conditions dans lesquelles une notification d'alerte est envoyée à un destinataire prédéfini via un périphérique spécifié.

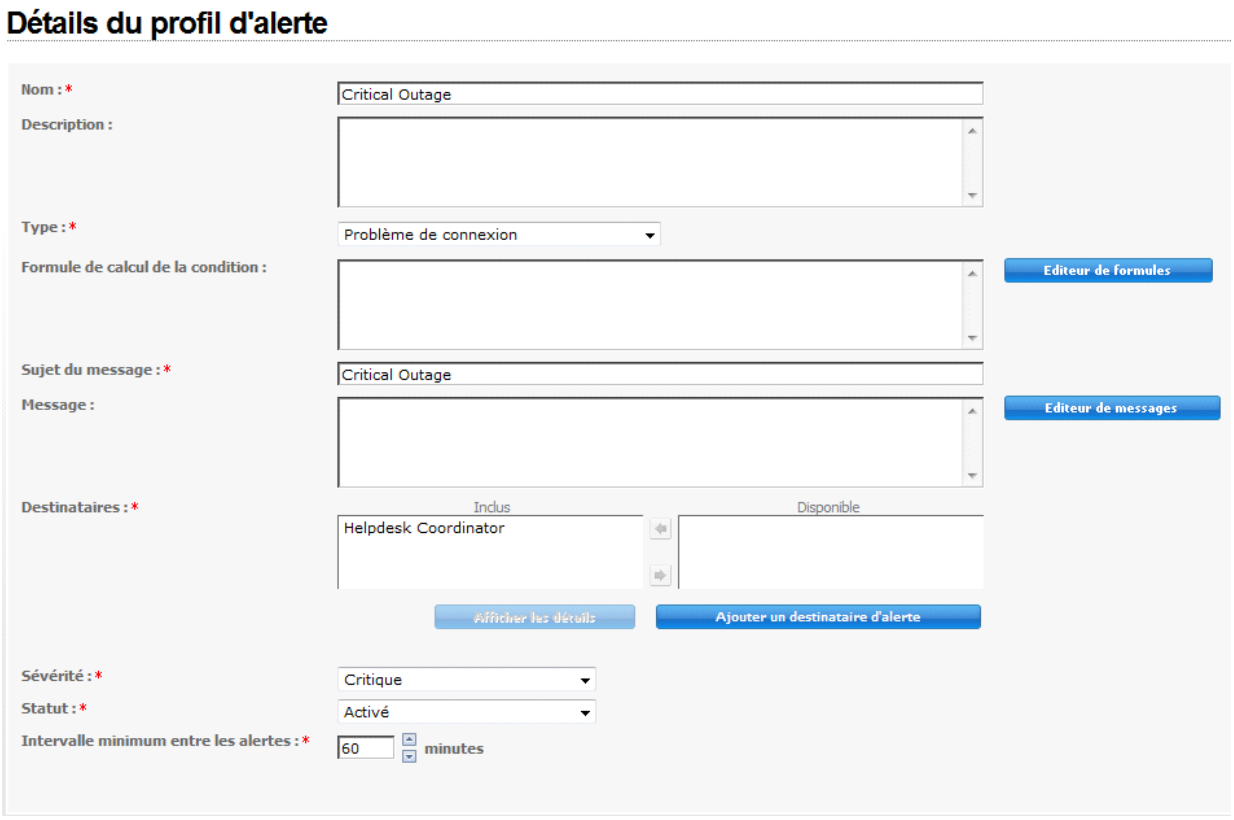
Avant de créer tout profil, il est utile d'examiner les conditions qui sont suffisamment importantes pour exiger une notification et également l'identité des destinataires voulus. Il importe de définir des profils qui peuvent aider le gestionnaire de contrats et l'administrateur système à gérer efficacement les problèmes et les difficultés liées aux services. Toutes les notifications doivent être spécifiques de manière à avertir les ressources adéquates pour résoudre les problèmes les plus urgents afin d'éviter une situation d'infraction.

Lors de l'ajout d'alertes de niveau de service, plusieurs champs peuvent être utilisés pour déterminer le statut de la métrique et pour savoir si une condition d'alerte doit être déclenchée ou non. Tous les détails de métrique standard peuvent servir comme critères de filtre ou de condition (p. ex., le contrat, le service, les noms de métrique, les cibles, les périodes d'application, les résultats de niveau de service, etc.). Des alertes peuvent également être émises en fonction des prédictions de niveau de service, auquel cas certaines valeurs supplémentaires, comme la déviation cas optimiste/pessimiste, sont fournies. En outre, des niveaux de service peuvent être estimés. Ceux-ci sont extrêmement utiles pour décider si une violation de niveau de service peut être récupérée ou non, au cours d'une période donnée.

Les alertes basées sur le journal système et les sections générales de CA Business Service Insight sont d'autres fonctionnalités extrêmement utiles et permettent d'envoyer une alerte selon toute action se produisant dans le système (consignée dans le journal du système). Presque toutes les actions étant journalisées, les profils d'alerte jouissent d'un périmètre très vaste. Les profils d'alerte communs du journal système se composent comme suit :

- Notification d'un adaptateur exécutant une alerte pour l'expert en sources de données.
- Toute condition d'erreur consignée dans le journal, toute alerte envoyée par courriel à l'administrateur système.
- Les erreurs personnalisées créées par la logique applicative concernant des conditions spécifiques des sources de données, peuvent générer une alerte par courriel pour le groupe responsable du service dans l'organisation et le gestionnaire de contrats. Autrement dit, si un incident à haute priorité n'a pas été résolu dans le délai imparti, une alerte est envoyée au centre d'assistance pour faire remonter le problème dans la hiérarchie.
- Des tentatives de connexion non valides répétées peuvent être signalées par courriel à l'administrateur système pour qu'il détermine si quelqu'un essaye de pénétrer sans autorisation dans le système.
- Les nouvelles entrées de conversion provenant de l'adaptateur (autrement dit les nouvelles entrées de ressource détectées dans une source de données, qui peuvent nécessiter une conversion dans le système pour permettre de calculer correctement le SLA.

La fenêtre suivante, Détails du profil d'alerte, affiche la configuration d'un profil d'alerte personnalisé documentant un type d'événement personnalisé des événements ouverts de ticket du centre d'assistance. Si les critères sont remplis, une alerte critique est envoyée au coordinateur du centre d'assistance pour garantir la gestion efficace de ce ticket. Ce peut être utile lorsqu'une application est sous examen et requiert un niveau d'attention particulier.


Détails du profil d'alerte
Nom : *
Critical Outage
Description :
Type : *
Problème de connexion
Formule de calcul de la condition :
Editeur de formules
Sujet du message : *
Critical Outage
Message :
Editeur de messages
Destinataires : *
Inclus
Disponible
Helpdesk Coordinator
Afficher les détails
Ajouter un destinataire d'alerte
Sévérité : *
Critique
Statut : *
Activé
Intervalle minimum entre les alertes : *
60
minutes

Voici un exemple d'alerte de violation de niveau de service, basée sur la possibilité d'encore satisfaire ou non le SLA, compte tenu du niveau de service actuel et du temps restant pour la période de suivi de la métrique.

Exemple : alerte de violation de SLA réversible

Cette alerte se déclenche lorsqu'un objectif (métrique) est actuellement transgressé, mais que la violation peut être évitée moyennant un niveau de service plus convenable dès à présent, dans la mesure où le scénario de cas optimiste indique une conformité avec l'objectif.

Type : prévision du niveau de service

Formule de condition :

#Deviation>0 And #BestCaseDeviation<=0 And #BestCaseServiceLevel<>#WorstCaseServiceLevel And #ClusterItem = ""

Message :

Notez que l'objectif #Metric du contrat #Contract est actuellement transgressé, mais que cette violation est réversible.

Lorsque la cible de niveau de service est #Target, les niveaux de service prévus sont les suivants :

- Niveau de service jusqu´à maintenant : #ServiceLevel (Déviation : #Deviation%).
- Niveau de service dans le pire scénario : #WorstCaseServiceLevel (Déviation : #WorstCaseDeviation%).
- Niveau de service dans le meilleur scénario : #BestCaseServiceLevel (Déviation : #BestCaseDeviation%).

Importation des profils d'alerte administratifs

L'option Transfert de contenu du menu Administration vous permet d'importer ou d'exporter des entités vers et à partir de la base de données système à des fins administratives. Il est utile d'importer certains profils d'alerte prédéfinis. Ces profils d'alerte apportent à l'administrateur système des alertes telles que les suivantes :

- **Notification du statut de l'adaptateur** : l'alerte signale tout changement de statut de l'adaptateur.
- **Suivi des connexions incorrectes** : notifie les événements de connexion erronée à CA Business Service Insight.
- **Notification de service CA Business Service Insight** : notifie l'initialisation de service interne.
- **Traitement des pénalités** : l'alerte signale les résultats du traitement de la pénalité.
- **Traitement du niveau de service** : l'alerte signale les résultats du traitement du niveau de service.
- **Notification d'erreur système** : l'alerte signale les messages d'erreur du système.

Procédez comme suit :

1. A partir des fichiers d'installation du dossier Additional Files, copiez le fichier SupportAlertProfiles.xml sur le serveur d'applications dans le dossier Program Files\CA\Insight\Export\Out. Remarque : Si vous n'avez pas reçu ce fichier, vous pouvez contacter le support de CA pour le demander.
2. Dans le menu Administration, cliquez sur Transfert de contenu, Importer. Dans la boîte de dialogue Importer, sélectionnez Ignorer et continuer pour l'option En cas de collision. Sélectionnez le fichier SupportAlertProfiles dans la liste déroulante Fichier à importer.
3. Cliquez sur Importer. Les profils sont importés.

Chapitre 4 : Installation et déploiement

Ce chapitre traite des sujets suivants :

Introduction (page 228)
Préparations (page 230)
Installation (page 232)

Introduction

Une fois la phase de configuration terminée et les calculs système vérifiés correctement, le système est prêt à être déployé dans l'environnement de production. Ce chapitre vise à couvrir toutes les étapes qui peuvent être nécessaires pendant cette phase. Ces étapes peuvent varier d'une installation à l'autre ; toutefois, il convient de vérifier les éléments suivants pour vous assurer que toutes les préparations requises ont été réalisées afin d'intégrer le système à un environnement de production en direct.

- Le matériel et le logiciel de production sont disponibles et prêts à l'installation.
- Le serveur de messagerie est configuré pour prendre en charge l'envoi de courriel externe/interne.
- L'agent FTP existe (si requis pour les transferts de fichier de données).
- Les flux de source de données sont disponibles et fonctionnent.

L'installation comporte deux phases. Pendant la première phase, la phase d'installation, l'administrateur système doit installer et intégrer les composants de CA Business Service Insight à l'environnement de production. Pendant la deuxième phase, la fonctionnalité du système est vérifiée et le système doit être mis sous contrôle pour s'assurer que les processus système et les fonctionnalités d'interface utilisateur fonctionnent tous correctement.

Pendant le processus d'installation, il peut être requis de travailler sur les serveurs de production via des outils de connexion distants qui doivent être installés dans l'application et sur les serveurs Web pour assurer que l'installation est complète et sûre.

L'installation de la base de données Oracle doit être réalisée par un spécialiste DBA Oracle si possible, pour garantir une configuration correcte et conforme aux exigences d'entreprise en vigueur. La création de base de données peut être confiée à l'administrateur système, moyennant recours à l'utilitaire de création de données fourni avec l'installation de CA Business Service Insight. Dans ce cas, l'installation doit être vérifiée par le DBA pour s'assurer qu'elle a été correctement accomplie. Si le système a déjà été configuré sur un système de développement et a besoin d'être migré, il est nécessaire de transférer le contenu de la base de données sur cette nouvelle base de données de production.

Une fois que le système complet a été transféré dans le nouvel environnement de production, il est parfois recommandé de déposer toutes les données calculées et brutes stockées dans le système et de charger de nouveau le système à partir de zéro (en gardant à la place l'ensemble du catalogue de services, des ressources et des conversions, bien entendu). C'est une excellente façon de tester le fonctionnement de l'ensemble du système dans l'environnement de production.

Pour récapituler, les étapes suivantes sont incluses dans cette phase :

- Installez et connectez le système.
- Chargez la base de données système configurée, le cas échéant.
- Intégrez les connexions pour le courriel, Exchange, les SMS, etc.
- Testez la fonctionnalité et ajustez les performances.
- Installez, configurez et testez les connexions distantes.
- Ré-initialisez le système pour qu'il soit prêt à opérer en direct.
- Activez les adaptateurs de données pour commencer à interroger les sources de données et à charger des données brutes.
- Démarrez le moteur de CA Business Service Insight pour permettre le début des calculs.
- Remplissez toute la documentation nécessaire, notamment concernant l'administration système, la base de données et les autres procédures de maintenance.
- Assurez que toutes les formations requises sont fournies aux utilisateurs.

Préparations

Afin d'assurer l'efficacité de l'installation, il importe de préparer certaines choses à l'avance :

1. **L'exportation de base de données de l'environnement de développement**. Cette exportation de base de données doit s'accomplir en utilisant des scripts CA approuvés qui créent un fichier d'extraction (vidage) afin de pouvoir la réimporter dans le logiciel du système de production.

   Remarque : Il est important que l'exportation soit accomplie par un utilisateur de la base de données jouissant de droits suffisants et que ce même utilisateur soit accessible lors de l'importation dans la base de données. A cette fin, vous pouvez utiliser soit le compte **oblicore**, puisque celui-ci est toujours créé sur chaque système, soit le compte **sys**. Toutefois, il convient de s'assurer que le mot de passe **sys** est disponible sur la base de données cible pour lancer le processus d'importation.

2. **Scripts de base de données** : si le DBA souhaite modifier les scripts de création de base de données, ceux-ci doivent être vérifiés d'avance par un DBA de CA. Ces scripts doivent être vérifiés et prêts d'avance pour permettre une installation sans écueils. Il est fréquent que différentes organisations aient des commentaires et des modifications qui doivent être approuvés par l'administrateur de base de données de CA pour assurer que la configuration de la base de données respecte les stratégies locales.

   - **Connectivité et accessibilité du serveur** : les applications comme les serveurs Web doivent avoir un client de connexion distante installé, afin de prendre en charge l'accessibilité externe (si l'accès physique n'est pas possible au moment de l'installation ou à l'avenir pour les questions d'assistance). L'accessibilité externe est également importante pour permettre d'assurer le suivi du système pendant une période de règlement. Si l'accès distant externe est exigé, l'organisation pourra être plus longue en raison des implications de sécurité supplémentaires ; il faudra donc résoudre cela à l'avance.
     Des connexions distantes peuvent être établies à l'aide de l'un des outils suivants :
   - PcAnyWhere
   - Hôte proxy/Maître
   - Microsoft Terminal Server
   - Bureau à distance
   - VNC

- Tout autre outil de connexion à distance pris en charge par l'organisation.

Remarques :

1. Microsoft Terminal Server se connecte au serveur en simulant une connexion de serveur, contrairement à PcAnyWhere, à VNC, au proxy et à d'autres outils.
2. Certains logiciels Oracle ne peuvent pas être installés via Microsoft Terminal Server et doivent être installés localement.

3. **Accessibilité de la source de données** : la source de données doit être accessible pour toute manipulation manuelle et la connexion ODBC doit être configurée pour permettre aux adaptateurs de se connecter à la source de données.
4. **Sécurité du serveur** : dans l'application comme dans le serveur Web, un profil d'utilisateur doté de droits d'administrateur local est requis. Si la plate-forme de base de données est un système d'exploitation Windows standard, un profil d'utilisateur doté de droits d'administrateur local est également requis. Pour les plates-formes Unix, l'administrateur de base de données doit avoir les droits appropriés pour créer la base de données sur le serveur.
5. **Accès à Internet** : celui-ci permet à l'**administrateur système** de se connecter à Internet pour toute mise à jour du système d'exploitation ou des applications, le cas échéant.
6. **Un PC de bureau d'utilisateur standard, pour tester la fonctionnalité de l'application**. Remarquez que l'application exige l'installation de certains contrôles ActiveX, qui sont souvent verrouillés par les stratégies de sécurité organisationnelles.
7. **Planification de formation pour le personnel concerné** : formation préliminaire du personnel pour lui permettre d'accomplir la formation d'acceptation utilisateur et de commencer à utiliser le système.

Installation

Le processus d'installation est décrit en détail dans le Manuel d'installation et inclut les activités suivantes :

1. **Installation de la base de données** :

   La responsabilité de l'installation de la base de données incombe à l'expert en sources de données ou au DBA Oracle et, dans certaines circonstances spéciales, doit s'accomplir sous la direction d'un ingénieur CA. L'installation de la base de données inclut les étapes suivantes :

   - Installation d'Oracle sur le serveur de base de données.
   - Installation du patch Oracle sur le serveur de base de données (le cas échéant - toujours utiliser les dernières Service Releases du logiciel de base de données Oracle).
   - Installation du client Oracle sur le serveur d'applications/Web.
   - Installation du patch de client Oracle sur le serveur d'applications/Web (le cas échéant - toujours s'assurer que cette version correspond au serveur).

2. **Installation** de l'application CA Business Service Insight :

   L'installation de l'application est accomplie par l'administrateur système. Si l'installation a déjà été réalisée dans l'environnement de test (pendant la phase de configuration) au cours de l'intégration et du test d'adaptateur, seul un état d'initialisation ou une importation de base de données est requise. L'installation dans l'environnement de production peut alors être requise.

   L'installation de l'application comporte les étapes suivantes :

   - Création de la base de données de CA Business Service Insight.
   - Installation du serveur d'applications.
   - Installation du serveur Web.
   - Installation de l'adaptateur.

Installez toujours la dernière Service Release CA sur les serveurs d'applications/Web. Pendant l'installation de l'application et des Service Releases, les scripts SQL sont inclus pour assurer que la base de données est mise à niveau selon la structure correcte de la version. Ces scripts sont stockés dans le répertoire d'installation de l'application, dans l'éventualité d'une mise à jour ultérieure de la base de données. (Par exemple, si vous avez besoin d'importer la base de données à partir d'une sauvegarde créée sur la base d'une Service Release précédente. Dans ce cas, vous devez exécuter le tout dernier script de mise à jour d'une Service Release pour garantir que la structure de la base de données correspond aux composants binaires installés dans le cadre de cette Service Release.)

Vous pouvez également installer des outils de prise en charge supplémentaires, comme le développeur de PLSQL ou l'utilitaire SQL alternatif pour contribuer à toute manipulation de données de bas niveau pouvant être exigée.

Importation/exportation entre environnements (expert en sources de données)

Cette situation couvre la migration d'adaptateurs qui ont été configurés et testés dans un environnement de développement/test vers un environnement de production ou en direct. Il est entendu que l'environnement de production a déjà été installé avec une installation de CA Business Service Insight standard et que la base de données est présente.

Le processus se compose comme suit :

Pour exporter la base de données et l'importer dans le nouvel environnement

1. Arrêtez tout travail sur le système de développement et choisissez un point logique dans le système pour accomplir une exportation que vous pouvez utiliser pour le système de production. Arrêtez tous les composants de service de CA Business Service Insight et les composants COM+ de CA Business Service Insight. Effectuez l'exportation à l'aide des scripts d'exportation de système standard de CA Business Service Insight. Le cas échéant, contactez le service d'assistance de CA.
2. Prenez la copie de l'extrait de base de données et placez-la sur le serveur de production pour importation. Notez que les versions Oracle des bases de données de développement et de production doivent correspondre. Importez la base de données en utilisant les scripts d'importation de système standard (fournis avec les scripts d'exportation)
3. Une fois l'opération terminée, repérez les éventuels problèmes d'importation. En l'absence de problème, assurez-vous d'exécuter les scripts SQL de la dernière Service Release (si nécessaire).
4. Exécutez le raccourci Petit assistant afin de configurer la base de données pour les paramètres du nouveau système de production.
5. Démarrez tous les composants de service de CA Business Service Insight et connectez-vous au système de production pour confirmer que le système a été importé correctement.

Pour migrer les adaptateurs

1. Installez l'adaptateur à l'aide de l'utilitaire AdapterManager avec les paramètres similaires à l'adaptateur que vous importez (assurez-vous que le nom est exactement le même afin que l'étape suivante fonctionne correctement).
2. Copiez le fichier de configuration de l'adaptateur à partir du système de développement vers le nouveau dossier d'adaptateur du système de production. Remplacez la configuration par défaut fournie. Veillez à ce que le fichier soit écrasé. Si ce n'est pas le cas, cela signifie que le nom n'est pas le même, ce qui cause des problèmes lors de l'exécution.
3. Mettez à jour le fichier de configuration de l'adaptateur. La communication avec le serveur CA Business Service Insight et à la source de données doit être mise à jour pour s'adapter au nouvel environnement. La section OblicoreInterface doit être mise à jour avec le port d'adaptateur correct. La partie DataSourceInterface doit être mise à jour avec la valeur ConnectionString, le modèle de nom de fichier ou le chemin d'accès adéquat, le cas échéant.
4. Assurez-vous que tous les DSN (Data Source Name, Nom de source de données) ODBC sont configurés et fonctionnent sur le nouvel ordinateur.
5. Testez la connectivité de l'adaptateur.
6. Testez l'exécution de l'adaptateur.
7. **Conversions** : s'il existe des scripts de conversion, ceux-ci doivent être activés. Vérifiez qu'ils sont synchronisés avec l'adaptateur et que les conversions fonctionnent comme prévu. Si la conversion manuelle est utilisée et que celle-ci n'a pas été terminée au préalable, il convient d'accomplir d'autres conversions à ce stade.
8. **Planification de l'adaptateur** : planifiez l'adaptateur pour qu'il s'exécute comme prévu. Si l'adaptateur a été défini comme application en mode Exécuter une fois, il peut être planifié à l'aide du Planificateur de Windows. Pour l'afficher, allez dans Panneau de configuration > Tâches planifiées. Consultez la section Modes d'exécution de l'adaptateur (page 101) pour plus d'informations.

Intégration - Installation du serveur de messagerie (Administrateur système)

Pour activer les fonctionnalités de notification du système, il est nécessaire de savoir quel serveur de messagerie et quelle boîte aux lettres sont utilisés pour envoyer les courriels de CA Business Service Insight. Ce serveur de messagerie doit permettre le relais du courriel, puisqu'il sera envoyé au format SMTP depuis les serveurs CA Business Service Insight vers ce serveur de messagerie, à l'aide du compte spécifié. Après avoir terminé l'installation de serveur de messagerie, vous pouvez utiliser des fonctionnalités de courriel dans les fonctions d'alertes, de rapports de détection de service et d'approbation de contrat dans CA Business Service Insight

Cliquez sur le menu Administrateur et sélectionnez Paramètres du site, Alertes. Configurez les définitions de courriel dans la section de paramètres d'alerte, notamment le serveur de messagerie, l'adresse d'envoi et le nom de l'expéditeur, avec des informations de fournisseur de SMS pour l'utilisation de passerelles de SMS.

Le test de l'intégration implique en partie de vérifier que le serveur d'applications peut atteindre le serveur de messagerie de l'organisation et l'utiliser pour envoyer des alertes CA Business Service Insight.

Pour tester la connectivité entre ces serveurs :

Entrez une invite de ligne de commande sur le serveur d'applications comme suit :

ORGANISATION-MAIL est généralement le serveur de messagerie défini sur votre client de messagerie. Contactez votre administrateur système afin de recevoir ce paramètre si vous avez un doute.

Remarque : Le nom ORGANISATION-MAIL est un espace réservé pour le serveur de messagerie de votre organisation. Vous devez remplacer cet espace réservé par le nom/l'adresse de votre serveur de messagerie.

Pour vérifier le serveur de messagerie client Outlook :

1. Dans Microsoft Outlook, allez dans Outils > Comptes de messagerie et sélectionnez Afficher ou modifier les comptes de messagerie existants.
2. Cliquez sur Modifier
3. Copiez le serveur Microsoft Exchange qui est le serveur de messagerie de l'organisation**.**

   Si vous obtenez une réponse du serveur avec cette commande, la connexion est réussie. La réponse doit être similaire à ce qui suit :

   Si vous recevez un autre message, cela indique l'absence de connectivité entre les deux serveurs. Vous devrez alors vous renseigner auprès de l'administrateur système.

Définition des préférences système (administrateur système)

Le processus de définition des préférences système inclut l'application de valeurs pertinentes aux variables système. A partir du menu Administration, cliquez sur Paramètres du site, Moteurs pour ouvrir la boîte de dialogue Préférences du moteur.

Pour obtenir des détails quant aux diverses recommandations de paramètre, consultez le Manuel d'administration.

Paramètres de sécurité (administrateur système)

Les paramètres de sécurité incluent la création d'utilisateurs, de groupes d'utilisateurs et de rôles. Par défaut, tous les utilisateurs sont associés à l'organisation spécifiée pendant le processus d'installation de l'application. Toutefois, il est également possible de créer des organisations supplémentaires, le cas échéant.

La plupart des définitions requises ont déjà été réalisées pendant la phase de configuration. Par conséquent, généralement, seul un simple ajustement est requis pour définir les paramètres supplémentaires qui ont pu être identifiés depuis lors.

Pour de plus amples détails à propos des paramètres de sécurité, consultez l'aide en ligne ou le Manuel de l'administrateur.

Spécification des paramètres pour la synchronisation SSA

Lorsque vous utilisez CA Spectrum Service Assurance (SSA) afin de détecter des services pour CA Business Service Insight, vous pouvez configurer des paramètres de manière à activer la synchronisation automatisée. Les fonctionnalités d'automatisation vous permettent de garder à jour la liste des services et les données de service.

Remarque : Vous devez avoir accès à l'API RESTful dans SSA pour modifier ces paramètres.

Procédez comme suit :

1. Dans le menu Administrateur, cliquez sur Paramètres du site, Paramètres SSA.

   La boîte de dialogue Paramètres SSA s'ouvre.

2. Introduisez les informations suivantes dans la zone Authentification d'utilisateur SSA :

   URL du serveur

   Spécifie l'URL du serveur SSA cible.

   ID de l'utilisateur

   Spécifie l'ID de l'utilisateur du serveur SSA.

   Mot de passe

   Spécifie le mot de passe correspondant à l'ID d'utilisateur du serveur SSA.

3. Introduisez les informations suivantes dans la zone Options de synchronisation :

   Synchronisation automatique

   Indique que vous souhaitez une synchronisation automatique, à la fréquence définie ci-dessous.

   Définition d'une fréquence de synchronisation

   Spécifie la fréquence à laquelle rechercher de nouveaux services. Vous pouvez indiquer une valeur en heures ou en jours.

   Synchronisation manuelle

   Permet la synchronisation manuelle à partir de la boîte de dialogue.

4. Introduisez les informations suivantes dans la zone Options de données par défaut :

   Définir les services par défaut sur :

   Permet de définir la valeur par défaut sur Géré ou Non géré pour les services détectés.

   Activer le calcul des métriques de comparaison

   Permet de déterminer l'action par défaut pour définir des métriques de comparaison pour les services SSA.

5. Cliquez sur OK.

   Vos paramètres SSA sont activés.

Paramètres de partage global

Les paramètres de partage global permettent d'établir un lien avec Cloud Commons. Une fois le lien établi, vous pouvez partager et récupérer des données de comparaison d'homologues, qui s'affichent dans l'ensemble de l'interface utilisateur, partout où des données de comparaison sont utilisées.

Les données concernant votre société sont utilisées pour quelques calculs statistiques sur Cloud Commons, mais sont gardées confidentielles. Cliquez sur la politique de confidentialité de CA dans la zone Sélectionner l'option de partage pour plus informations.

Procédez comme suit :

1. Cliquez sur Paramètres du site, Partage global dans le menu Administration.

   La boîte de dialogue Paramètres de partage global s'ouvre.

2. Introduisez les informations suivantes dans la zone Informations démographiques pour la comparaison de données homologues :

   Secteur

   Spécifie la catégorie de votre secteur. Sélectionnez un nom de secteur dans le menu déroulant.

   Taille de la société

   Spécifie la taille de votre société. Sélectionnez une taille dans le menu déroulant.

   Chiffre d'affaires

   Spécifie le chiffre d'affaires annuel de votre société.

   Pays

   Spécifie le pays où se situe le siège de votre société.

   Code postal

   Spécifie le code postal du siège de votre société.

3. Introduisez les informations suivantes dans la zone Enregistrement de la clé API :

   Nom de l'organisation

   Spécifie le nom de votre organisation.

   Courriel de l'administrateur

   Spécifie l'adresse électronique de l'administrateur pour les tâches de partage global.

Confirmer le courriel de l'administrateur

Vérifie l'adresse électronique introduite dans le champ précédent.

4. Dans la zone Sélectionner l'option de partage, parcourez l'accord pour le partage global et sélectionnez l'une des options :

   Partage global : Activer

   Active l'option de partage global pour vos services. Cette option permet de marquer les différents services pour le partage à l'aide du bouton Action du service dans la page Gestion et détection de services.

   Partage global : Désactiver

   Désactive le partage global pour vos services. Lorsque le partage global est désactivé, vous n'avez pas la possibilité de partager différents services dans la page Gestion et détection de services.

5. Cliquez sur Enregistrer pour le premier enregistrement ou sur Enregistrer pour mettre à jour des modifications.

   Vos paramètres de partage sont activés.

Chapitre 5 : Gestion et détection de services

Ce chapitre traite des sujets suivants :

Détection de services (page 243)

Détection de services

Fonctionnement de la gestion et de la détection de services

Le gestionnaire de dossiers de modèles/gestionnaire de détection de services effectue des tâches administratives sur les services détectés à l'aide de fonctionnalités de la page Gestion et détection de services.

Après avoir terminé les tâches préliminaires pour détecter vos services, vous pouvez utilisez des fonctionnalités de la page Gestion et détection de services pour réaliser les opérations suivantes :

- Comprendre les catégories de service (page 245)
- Définir la visibilité et le périmètre de vos services (page 248)
- Classer un service sélectionné (page 253)
- Utiliser des options d'attribut et de métadonnées (page 254)
- Gérer vos services et les activer pour le partage (page 258)
- Détecter de nouveaux services et les synchroniser avec CA Spectrum Service Assurance (SSA) (page 262)

Cliquez sur Conception, Détection de service dans le menu principal de CA Business Service Insight pour démarrer.

Présentation des catégories de services

Comme vous travaillez avec vos services, vous pouvez les classer pour activer les fonctionnalités de comparaison et d'analyse disponibles dans CA Business Service Insight. La catégorisation des services commencer lorsque vous détectez des services à l'aide de CA Service Spectrum Assurance (SSA) et se précise au fur et à mesure que vous travaillez avec des fonctionnalités dans la page Gestion et détection de services.

Vous pouvez associer une catégorie de service à un service spécifique. La manière dont vous travaillez avec des options de catégorisation affecte les fonctionnalités d'affichage, de filtrage et de comparaison des services.

Les catégories de service suivantes sont fournies et s'affichent par défaut dans le volet Afficher les filtres.

- Sauvegarde et récupération
- Aide à la décision
- Calculer
- Gestion de la relation client
- Base de données
- Commerce électronique
- Courriel
- Planification des ressources d'entreprise
- Gestion IT
- Plate-forme
- Gestion de projet et de portefeuilles

Consultez la Feuille de calcul des catégories de services (page 245) pour plus d'informations sur la définition de chaque catégorie.

Feuille de calcul des catégories de services

Lorsque vous catégorisez vos services, les informations de cette feuille de calcul peuvent vous aider à comprendre la signification des différentes catégories. Ces mêmes informations sont affichées dans la boîte de dialogue Catégorisation du service, dans une fenêtre contextuelle qui s'affiche lorsque vous pointez un nom de catégorie avec la souris.

Nom de catégorie	Les services de cette catégorie doivent inclure les fonctionnalités suivantes.
Sauvegarde et récupération	■ Performances de stockage ■ Mise à l'échelle automatique ■ Prise en charge des options régionales ■ Prise en charge des systèmes de fichiers ou des multiprotocoles ■ Option de stockage hors ligne ■ Sauvegarde dans le Cloud
Aide à la décision	■ Entreposage de données ■ Rapports de ventes ■ Rapports de données ■ Génération de rapports ad hoc ■ Intégration SalesForce ■ Prise en charge d'applications d'entreprise
Calcul	■ Instance de création ■ Instance de mise à l'échelle ■ Instance persistante ■ Intégration de service de stockage ■ Instance de répartition de la charge ■ Instance de moniteur ■ Intégration de service de base de données ■ Capacité de calcul ■ Capacité réseau ■ Performances de stockage

Gestion des relations client	■ Gestion des prospects ■ Gestion des services ■ Gestion des contacts ■ Intégration d'un système de messagerie ■ Analyse ■ Prévision ■ Suivi d'opportunités ■ Accès mobile ■ Magasin de documents ■ Suivi d'activités ■ Automatisation ou flux de travaux ■ Gestion des campagnes
Base de données	■ Performances de stockage ■ Relationnelle ■ Non relationnelle ■ Hébergée sur Amazon ■ Hébergée sur Rackspace ■ Création ou gestion de d'instances Cloud
Commerce électronique	■ Traitement des commandes ■ Gestion d'inventaires ■ Exécution ■ Services de paiement ■ Génération de rapports ■ Gestion des catalogues ■ Gestion des clients ■ Recherche de catalogues ■ Intégrations

Messagerie	■ Envoi et réception de courriels ■ Carnet d'adresses ■ Calendrier ■ Commentaires ■ Répertoires ■ Intégration IM ■ Prise en charge de Mobile Push
Planification des ressources d'entreprise	■ Gestion financière ■ Planification financière ■ Gestion de la chaîne d'approvisionnement et des inventaires ■ Expédition et exécution ■ Gestion des ressources humaines ■ Analyse financière et génération de rapports
Gestion informatique	■ Surveillance et alertes ■ Equilibrage de la charge et mise à l'échelle ■ Gestion des serveurs physiques ■ Base ITIL ■ Automatisation ou génération de scripts ■ Déploiement cloud ■ Gestion des actifs ■ CMDB ■ Gestion des coûts

Plate-forme	■ Environnement de développement ■ Gestion des applications ■ Java ■ Ruby on Rails ■ .Net ■ Magasin de données SQL ■ Stockage persistant ■ Gestion de la capacité ■ Surveillance et alertes ■ Performances des applications
Gestion des projets et portefeuilles	■ Gestion financière ■ Gestion des tâches ■ Jalons ■ Gestion du temps ■ Stockage de fichiers ■ Panneaux de messages ■ Gestion des projets ■ Gestion des portefeuilles ■ Gestion des ressources ■ Gestion des demandes

Définition de la visibilité et du périmètre de vos services

Lorsque vous ouvrez d'abord la page Gestion et détection de services, tous les services détectés sont affichés. Pour affiner votre travail sur les services et accéder à des fonctionnalités plus précises de détection de service, utilisez les options suivantes de l'onglet Gestion des données et du partage :

Définition d'une vue des services (page 249)

Définition des options d'affichage des colonnes (page 250)

Choix d'une action pour un service sélectionné (page 252)

Définition d'une vue des services

Vous pouvez définir une vue des services pour travailler avec un sous-ensemble de tous vos services. Par exemple, vous pouvez afficher uniquement les services qui partagent des données de comparaison sur Cloud Commons lorsque vous sélectionnez la vue Services partageant des données de comparaison.

Procédez comme suit :

1. Cliquez sur Détection de service dans le menu Conception.

   La page Gestion et détection de services s'ouvre.

2. Cliquez sur l'onglet Gestion des données et du partage.

3. Cliquez sur le menu déroulant Afficher et sélectionnez une option pour afficher vos services dans une des catégories suivantes :

   Tous les services

   Affiche tous les services détectés sur votre système. Cette liste inclut les services détectés à partir de CA Spectrum Service Assurance (SSA), les services saisis manuellement, les services partageant des données de comparaison et les services pour lesquels l'option Comparaison de services est activée.

   Services SSA

   Affiche uniquement les services qui sont détectés à partir de CA Spectrum Service Assurance (SSA). Cette liste est dressée après avoir installé et configuré SSA et après avoir établi des connecteurs dans votre environnement.

   Services saisis manuellement

   Affiche uniquement les services que vous avez entrés manuellement à partir du bouton Ajouter des services.

   Services partageant un indice de mesure de service

   Affiche uniquement les services qui partagent des données de comparaison d'indice de mesure du service. Contactez votre administrateur pour discuter de la définition des paramètres de partage globaux pour votre compte d'utilisateur.

   Services incluant des comparaisons de services sur

   Affiche uniquement les services pour lesquels la comparaison de services est active. Pour activer ou désactiver la comparaison des services, utilisez le menu déroulant de la colonne Comparaison de services.

   Services gérés

Affiche uniquement les services définis sur l'état Géré.

Services non gérés

Affiche uniquement les services définis sur l'état Non géré.

Votre affichage s'actualise pour afficher les services dans la vue sélectionnée.

Définition des options d'affichage des colonnes

Vous pouvez personnaliser les colonnes qui s'affichent dans la table des services de la page Gestion et détection de services. Par exemple, si tous vos services sont gérés activement, vous pouvez supprimer la colonne Gérer, car elle vous est inutile.

Procédez comme suit :

1. Cliquez sur Détection de service dans le menu Conception.

   La page Gestion et détection de services s'ouvre.

2. Cliquez sur le bouton Personnaliser.

   La boîte de dialogue Paramètres d'affichage de la table s'ouvre.

3. Déplacez le nom de colonne que vous voulez inclure dans l'affichage vers la liste Colonnes visibles.

   Les colonnes disponibles sont les suivantes :

   Services

   Spécifie le nom du service.

   Catégorie

   Spécifie la catégorie de service que vous avez affectée au service.

Source

Spécifie la source du service. Par exemple, les services détectés via CA Spectrum Service Assurance (SSA) sont marqués SSA.

Métriques de comparaison de services

Indique si le service est défini pour la comparaison de services ou non. Vous utilisez ces paramètres sur la liste déroulante Action.

Gérer

Indique si le service est géré ou non. Un service est non géré s'il a été détecté, mais que ses fonctionnalités de métriques et de partage sont désactivées. Vous définissez un service comme Géré via la liste déroulante Action.

Nom du service partagé

Spécifie le nom pour les données de comparaison de services lorsqu'elles sont partagées sur Cloud Commons.

Partage de données

Indique si le partage de comparaison de services est actif pour le service ou non. Vous pouvez activer ou désactiver le partage via la liste déroulante Action.

Approbation requise

Indique si le nom de service a été soumis à Cloud Consortium, mais attend encore une approbation.

Date de détection

Affiche la date de détection du service.

4. Cliquez sur Enregistrer pour enregistrer vos paramètres et actualiser l'affichage de la table.

Choix d'une action pour un service sélectionné

Les options de la liste déroulante Action du service permettent de travailler avec un service sélectionné.

Procédez comme suit :

1. Cliquez sur Détection de service dans le menu Conception.

   La page Gestion et détection de services s'ouvre.

2. Cliquez sur l'onglet Gestion des données et du partage.

3. Cliquez sur le bouton Action du service.

   Les actions suivantes s'affichent dans le menu déroulant :

   Catégoriser/Modifier

   Catégorise un service sélectionné (page 253) pour activer les fonctionnalités de filtrage et de comparaison. Une fois un service catégorisé, cette option de menu change en **Modifier**.

   Activer les métriques de comparaison de services

   Active les calculs de métriques sur les services sélectionnés.

   Désactiver les métriques de comparaison de services

   Désactive les calculs de métriques sur les services sélectionnés.

   Activer le partage du service

   Active le partage de données de comparaison pour utilisation sur Cloud Commons.

   Désactiver le partage du service

   Désactive le partage de données de comparaison sur Cloud Commons.

   Configurer le service pour le partage

   La sélection de cette option permet d'utiliser des fonctionnalités de partage de données sur Cloud Commons.

   Gérer

   La sélection de cette option définit le service sur l'état Géré.

   Ne pas gérer

   La sélection de cette option définit le service sur l'état Non géré. Lorsqu'un service est non géré, les données de comparaison ne sont pas collectées pour ce service et le nom s'affiche dans une police rouge dans la liste des services.

Regérer

La sélection de cette option redéfinit l'état du service sur Géré.

Supprimer le service

La sélection de cette option supprime le service de CA Business Service Insight.

Une nouvelle boîte de dialogue s'ouvre ou les tables de services s'actualisent, selon votre sélection.

Catégorisation d'un service sélectionné

La catégorisation d'un service active des fonctionnalités de CA Business Service Insight qui vous permettent de recueillir des données et d'analyser les performances du service. Une fois un service catégorisé, vous pouvez filtrer la page Aperçu des services selon cette catégorie et comparer le service avec d'autres catégorisés de la même façon.

Procédez comme suit :

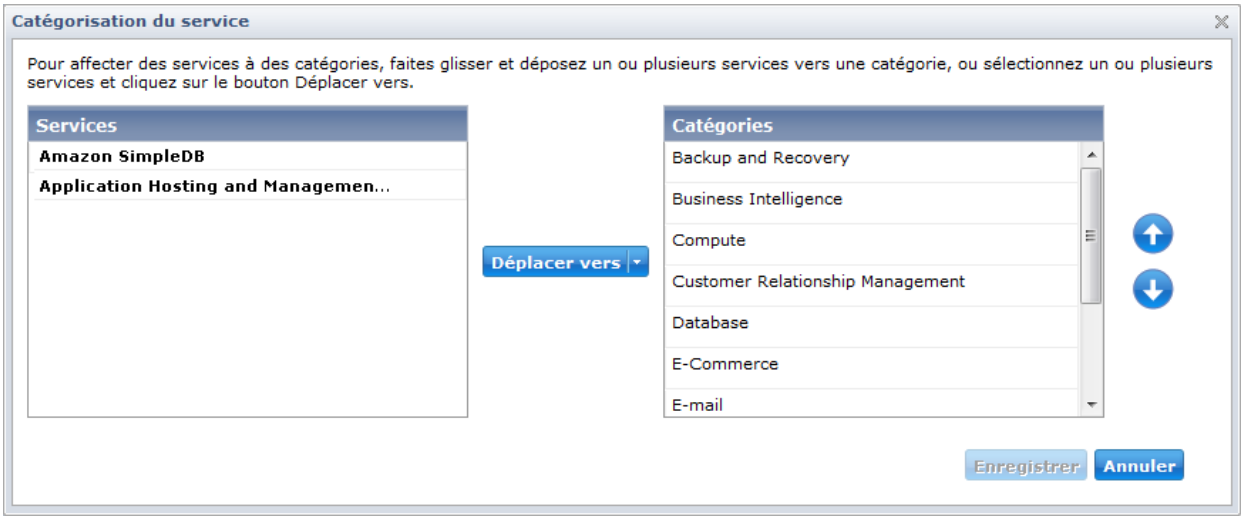
1. Cliquez sur Détection de service dans le menu Conception.

   La page Gestion et détection de services s'ouvre.

2. Sélectionnez la case à cocher en regard d'un ou de plusieurs services que vous voulez catégoriser, puis cliquez sur le bouton Action du service et sélectionnez Catégoriser.

   La boîte de dialogue Catégorisation du service s'ouvre.

   Vos services sélectionnés s'affichent dans le volet gauche. Les catégories de services disponibles s'affichent dans le volet droit.



3. Sélectionnez le service et faites-le glisser vers la catégorie ou sélectionnez un ou plusieurs noms de service et cliquez sur le bouton Déplacer vers pour sélectionner une catégorie.
4. Cliquez sur Enregistrer pour enregistrer vos sélections.

Remarque : Utilisez l'icône de flèche vers le haut ou vers le bas pour réorganiser les noms de catégorie.

Utilisation des options d'attribut et de métadonnées

Les options sous l'onglet Gestion des attributs et des métadonnées permettent de créer et de gérer les noms d'attribut. Les noms d'attribut s'affichent dans le volet Filtre de service de la page Aperçu des services. Ces options vous permettent de créer des filtres d'affichage personnalisés. Par défaut, les fonctionnalités Catégorie du service basées sur la structure de l'indice de mesure du service sont fournies.

Création d'un nom d'attribut (page 255)

Association d'un service à un attribut (page 256)

Gestion des attributs et des valeurs associées (page 257)

Création d'un nom d'attribut

Vous pouvez créer un nouveau nom d'attribut, puis spécifier une action pour celui-ci.

Procédez comme suit :

1. Cliquez sur Détection de service dans le menu Conception.

   La page Gestion et détection de services s'ouvre.

2. Cliquez sur l'onglet Gestion des attributs et des métadonnées, puis cliquez sur le bouton Attributs et valeurs.

   La boîte de dialogue Ajouter/gérer des attributs s'ouvre. La table affiche les informations suivantes :

   Nom de l'attribut

   Spécifie le nom d'attribut qui s'affiche dans la liste de filtre sur la page Aperçu des services.

   Modifier

   Permet modifier les paramètres d'attribut.

   Supprimer

   Supprime l'attribut.

   Statut

   Indique si l'attribut est actif ou inactif.

   Valeur

   Répertorie les valeurs associées à l'attribut.

3. Cliquez sur le bouton Ajouter un attribut.

   La boîte de dialogue Ajouter un attribut/des valeurs s'ouvre. Utilisez les options suivantes :

   Nom de l'attribut

   Spécifie le nom du nouvel attribut.

   Affecter des valeurs

   Permet d'entrer des valeurs spécifiques pertinentes pour la catégorie d'attribut. Par exemple, si vous créez un nom d'attribut Gestionnaires, vous pouvez créer des noms de gestionnaire pour les valeurs associées.

4. Cliquez sur Enregistrer.

   La boîte de dialogue se ferme et votre nouveau nom d'attribut s'affiche dans la colonne Nom de l'attribut.

Association d'un service à un attribut

Après avoir créé des noms d'attribut, vous pouvez associer des services aux attributs.

Procédez comme suit :

1. Cliquez sur Détection de service dans le menu Conception.

   La page Gestion et détection de services s'ouvre.

2. Cliquez sur l'onglet Gestion des attributs et des métadonnées.

3. Sélectionnez les services que vous voulez utiliser, puis cliquez sur le bouton Action du service et sélectionnez Affecter des attributs.

   La boîte de dialogue Associer des services aux attributs s'ouvre.

   Les services que vous avez sélectionnés s'affichent dans le volet gauche et vos attributs, dans le volet droit.

4. Sélectionnez et faites glisser le service vers la valeur d'attribut.

   Remarque : Développez l'arborescence des attributs pour voir les sous-catégories en dessous d'un attribut particulier.

5. Cliquez sur Enregistrer pour enregistrer vos sélections.

   La colonne dans l'affichage de table est actualisée pour afficher l'association entre le service et l'attribut.

Gestion des attributs et des valeurs associées

Après avoir créé des noms d'attribut, vous pouvez utiliser des options pour gérer des attributs à partir de la liste déroulante Action de la boîte de dialogue Ajouter/Gérer des attributs.

Procédez comme suit :

1. Cliquez sur Détection de service dans le menu Conception.

   La page Gestion et détection de services s'ouvre.

2. Cliquez sur l'onglet Gestion des attributs et des métadonnées, puis cliquez sur le bouton Attributs et valeurs.

   La boîte de dialogue Ajouter/Gérer des attributs et des valeurs s'ouvre.

3. Utilisez les options suivantes de la boîte de dialogue :

   Modifier

   Ouvre la boîte de dialogue Modifier les attributs/valeurs, qui permet de changer le nom et les valeurs associées.

   Supprimer

   Supprime l'attribut dans la ligne sélectionnée.

   Définir le statut sur actif

   Active l'attribut pour qu'il s'affiche dans la page Aperçu des services.

   Définir le statut sur inactif

   Désactive l'attribut pour qu'il ne s'affiche pas dans la page Aperçu des services.

4. Cliquez sur Fermer pour enregistrer vos modifications et retourner à la page Détection de service.

Gestion de vos services

Vous pouvez contrôler l'affichage de vos services sur la page Aperçu des services en activant ou en désactivant des fonctionnalités pour un service spécifique. Par exemple, vous pouvez choisir de collecter et d'afficher des données de comparaison interne, mais de désactiver l'affichage des données de comparaison d'homologues et de catégories. Vous pouvez cesser de gérer un service pour suspendre la collection de toutes les données de comparaison, etc.

Services gérés/non gérés

Lorsque vous détectez un service, vous pouvez choisir de définir le statut Géré. Un service géré est activé pour d'autres actions, comme la détection de métriques de comparaison ou le partage sur Cloud Commons. Cette première étape est requise avant de pouvoir utiliser d'autres fonctionnalités de comparaison. Le statut de gestion des services est reflété dans la colonne Gérer de votre liste de services.

Les services dont le statut Gérer est désactivé ne s'affichent pas dans la page Aperçu des services.

Paramètre de métriques de comparaison de services

Vous pouvez activer ou désactiver l'option de collecte de données de métriques de comparaison. Cette option est une action que vous pouvez appliquer à un ou plusieurs services à l'aide du menu Action sous l'onglet Gestion des données et du partage de la page Gestion et détection de services. Le statut pour cette fonctionnalité s'affiche dans la colonne Métriques de comparaison de services de la liste Services, à savoir Activer ou Désactiver.

Les services dont le statut des métriques de comparaison de services est défini sur Activer affichent des scores de métriques de comparaison dans la page Aperçu des services, ainsi que dans d'autres pages reprenant les scores de métriques.

Partage global

Vous pouvez activer le partage global pour tous vos services dans la boîte de dialogue Partage global à laquelle vous accédez lorsque vous cliquez sur Paramètres du site, Partage global dans le menu Administration. Le partage global doit être défini sur Activer afin que cette fonctionnalité soit active pour un service spécifique. Activez le paramètre de partage global si vous envisagez de partager largement vos données de service sur Cloud Commons. Si vous devez limiter le partage de données, utilisez le paramètre global afin de désactiver cette fonctionnalité pour tous les services. Pour utiliser des fonctionnalités de partage global, vous devez d'abord vous enregistrer sur Cloud Commons à l'aide des options de la boîte de dialogue Partage global.

Le statut de partage global pour un service s'affiche dans la colonne Partage de données de la liste Services. Cette colonne indique la mention Configuration requise pour les services qui n'ont pas encore été définis pour le partage global.

Partage de données

Lorsque le partage global est activé, vous pouvez activer cette fonctionnalité pour un service ou un groupe de services sélectionnés. Ce processus inclut la connexion à Cloud Commons pour partager vos données de comparaison. Si aucune correspondance n'est trouvée pour le service ou la catégorie sélectionnée, vous avez la possibilité d'ajouter votre service à Cloud Commons.

Le statut de partage de données pour un service s'affiche dans la colonne Partage de données de la liste Services, à savoir Activer ou Désactiver.

Les services dont le statut de partage de données est activé affichent des données de comparaison d'homologues et de catégories dans la page Aperçu des services, ainsi que dans d'autres pages reprenant des données de métriques.

Partage des données de comparaison sur Cloud Commons

Vous pouvez configurer un service local pour que vos données de comparaison soient partagées sur Cloud Commons. Lorsque vous choisissez cette option, vous pouvez commencer à accéder aux données de comparaison de catégories et d'homologues sur Cloud Commons et êtes autorisé à fournir des informations qui affectent les scores de comparaison auxquels d'autres peuvent accéder.

Remarque : Vous devez activer le partage global avant de pouvoir partager des données de comparaison sur Cloud Commons. Dans le menu Administration, cliquez sur Paramètres du site, Partage global pour ouvrir la boîte de dialogue Paramètres de partage global (page 241).

Procédez comme suit :

1. Cliquez sur Détection de service dans le menu Conception.

   La page Gestion et détection de services s'ouvre.

2. Cliquez sur l'onglet Gestion des données et du partage.

3. Sélectionnez le service pour lequel vous voulez partager des données, puis cliquez sur le bouton Action du service et sélectionnez Configurer le service pour le partage.

   Une nouvelle fenêtre s'ouvre avec les options de partage de données suivantes pour le service sélectionné :

   Correspondance trouvée sur Cloud Commons

   Si une correspondance est trouvée sur Cloud Commons pour ce service, un tableau récapitulatif répertorie le nom du service, la définition, les catégories et les fonctionnalités. Cliquez sur le bouton Partager les données.

   Correspondance non trouvée sur Cloud Commons

   Si aucune correspondance n'est trouvée sur Cloud Commons pour ce service, cliquez sur Parcourir pour rechercher dans Cloud Commons un service alternatif ou utilisez des options afin d'ajouter votre service à Cloud Commons (page 261).

4. Cliquez sur le bouton Partager les données si une correspondance a été trouvée. Cliquez sur Soumettre à Cloud Consortium pour entrer votre soumission de catégorie.

   La boîte de dialogue Partage de données se ferme.

Ajout de votre service à Cloud Commons

Lorsque vous commencez à partager des données de comparaison sur Cloud Commons, il se peut que vous ne trouviez pas de correspondance pour votre service. Vous pouvez alors rechercher dans Cloud Commons une correspondance alternative ou demander à ce que votre catégorisation soit ajoutée à Cloud Commons.

Procédez comme suit :

1. Utilisez la procédure pour le partage des données de comparaison sur Cloud Commons (page 260).
2. Si aucune correspondance n'est trouvée, une nouvelle fenêtre s'ouvre et vous permet de parcourir la liste des services Cloud Commons pour y repérer une correspondance. Si vous ne parvenez pas à identifier une correspondance, soumettez votre service pour qu'il soit inclus dans Cloud Commons en procédant comme suit.
3. Cliquez sur l'icône Ajouter un service pour ouvrir une nouvelle fenêtre qui vous propose les options suivantes :

   Création d'un alias de nom de service

   Spécifiez le nom de service que vous voulez soumettre.

   Sélection d'une catégorie de service et de fonctionnalités

   Utilisez la liste des catégories de service et des fonctionnalités pour sélectionner celles qui correspondent à votre nom de service.

   Description du service

   Entrez des informations descriptives supplémentaires pour expliquer la raison de votre soumission.
4. Cliquez sur Soumettre à la base de données globale des indices de mesure du service.

   Votre demande est soumise et vous recevez un peu plus tard une confirmation par courriel.

Ajout de services dans la page Gestion et détection de services

Bien que la plupart de vos services puissent être détectés automatiquement, vous pouvez également ajouter manuellement un service dans la page Gestion et détection de services.

Procédez comme suit :

1. Dans la page Gestion et détection de services, cliquez sur l'onglet Gestion des données et du partage.
2. Cliquez sur le menu déroulant Ajouter des services et sélectionnez Manuellement.

   La boîte de dialogue Détail du service s'ouvre.
3. Cliquez sur l'onglet Présentation et entrez les informations suivantes :

   Source (en lecture seule)

   Indique que la source du service a été entrée manuellement.

   Nom

   Spécifie le nom du service à afficher dans la table des services.

   Description (facultative)

   Spécifie la description du service

   Statut

   Permet de spécifier si le service est géré ou non.

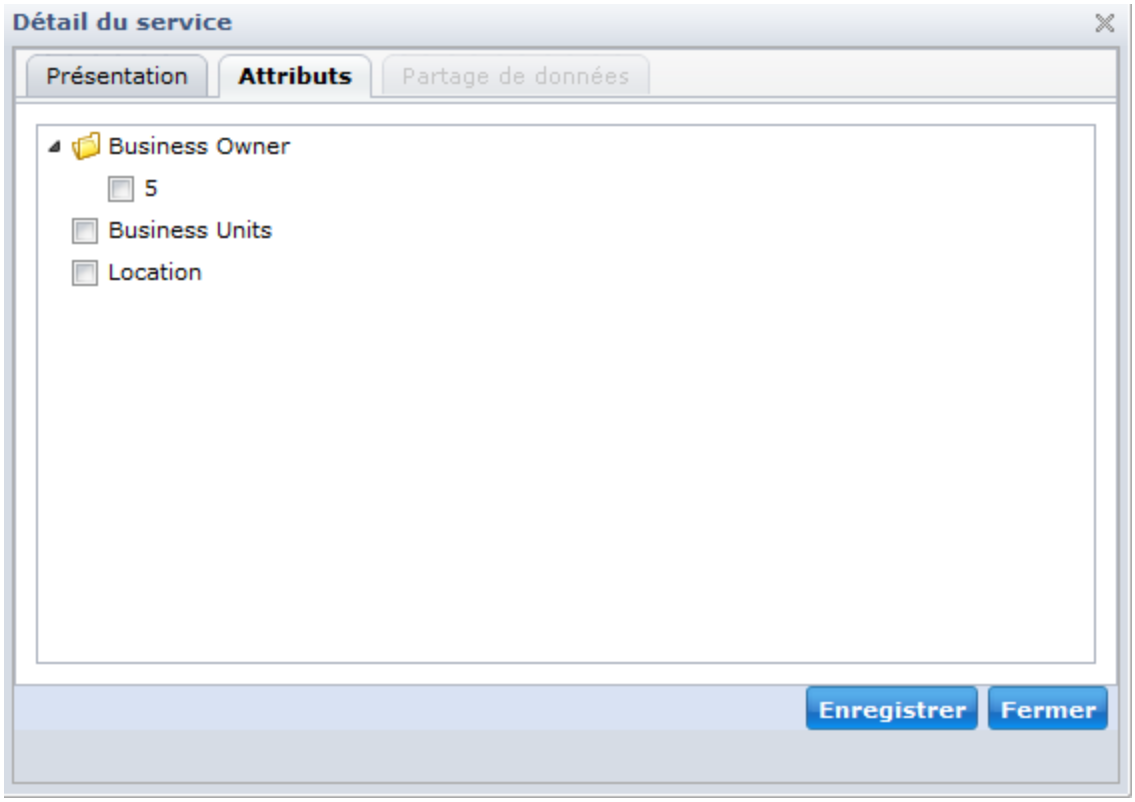
   Fournisseur

   Permet de sélectionner le fournisseur du service dans une liste déroulante.

   Activer le calcul des métriques de comparaison

   Permet d'activer ou de désactiver le calcul des métriques de comparaison.

4. Cliquez sur l'onglet Attributs pour associer un attribut existant au service. L'association permet alors de filtrer sur l'attribut dans les options de vue. Par exemple, pour ajouter l'attribut Emplacement New York au service, activez la case à cocher dans la liste Emplacement. Cliquez sur Enregistrer pour activer l'onglet Partage de données.



5. Cliquez sur l'onglet Partage de données et utilisez des options afin de partager des données pour ce service sur Cloud Commons.

Remarque : Vous pouvez également importer des données de service manuellement (page 266) à l'aide de la fonctionnalité InsertUtil.

Exportation et importation manuelles de données de service

CA Business Service Insight fournit un utilitaire qui vous permet d'exporter et d'importer des données de service à l'aide d'un script de ligne de commande. Il vous est possible d'exporter des données de service vers un fichier CSV ou XML, puis de modifier ce dernier en dehors de CA Business Service Insight. L'utilitaire vous permet également de formater des données de service à partir d'une autre source pour pouvoir ensuite les importer manuellement dans CA Business Service Insight.

L'utilitaire d'importation et d'exportation de service prend en charge les données suivantes :

Tous les attributs de service standard :

- Nom du service
- Description
- Nom du fournisseur
- Source du service
- Attribut de partage
- Attribut de gestion

Métadonnées de service - attributs personnalisés :

- Ajout de nouveaux attributs

Catégories de service et fonctionnalités

- Génère automatiquement des informations de catégorie si les informations de fonctionnalité sont fournies mais pas celles de catégorie.
- Génère automatiquement générez des informations de fonctionnalité si les informations de catégorie sont fournies mais pas celles de fonctionnalité.

Hiérarchie de service

- Des informations enfants peuvent être automatiquement déterminées à partir des informations parentes.

Exportation manuelle de données de service

Vous pouvez exporter des données de service manuellement vers un fichier CSV ou XML.

Le processus d'exportation utilise les commandes suivantes :

```
InsightUtil /exportservice [/http|/https] /server <hostname> /port <port#>
/format csv|xml /path <file_path> /key <key> /secret <secret>
```

InsightUtil

La commande commence avec le nom de l'utilitaire, à savoir InsightUtil.

/exportservice [/http|/https]

Indique que le service d'exportation utilise HTTP ou HTTPS.

/server <hostname>

Indique le nom d'hôte du serveur.

/port <port#>

Indique le numéro de port.

/format csv|xml

Indique le format du fichier d'exportation : CSV ou XML.

/path <file_path>

Indique le chemin pour distribuer le fichier d'exportation.

/key <key>

Indique la clé d'administrateur de CA Business Service Insight unique.

/secret <secret>

Indique le mot de passe d'administrateur de CA Business Service Insight unique.

Exemple : exportation des données de service vers un fichier CSV local

L'exemple ci-dessous illustre la commande d'exportation des données de service vers un fichier CSV local.

```
InsightUtil /exportservice /https /server name001 /port 8443
/format csv /path C:\MyFiles\file.csv /key mykey /secret
mypassword
```

Importation manuelle de données de service

Vous pouvez importer des données de service manuellement à partir d'un fichier CSV ou XML. Vous devez importer un fichier CSV ou XML au format défini pendant le processus d'exportation. Vous pouvez créer et utiliser un fichier de modèle de service à l'aide de la procédure d'exportation afin de vous assurer que vos données sont au bon format.

Le processus d'importation utilise les commandes suivantes :

```
InsightUtil /importservice [/http|/https] /server <hostname> /port <port#>
/format csv|xml /path <file_path> /key <key> /secret <secret>
```

InsightUtil

La commande commence avec le nom de l'utilitaire, à savoir InsightUtil.

/importservice [/http|/https]

Indique que le service d'importation utilise HTTP ou HTTPS.

/server <hostname>

Indique le nom d'hôte du serveur.

/port <port#>

Indique le numéro de port.

/format csv|xml

Indique le format du fichier d'importation : CSV ou XML.

Important : Lors de l'importation d'un fichier CSV, la première ligne (la ligne de colonne d'en-tête) doit être présente, faute de quoi le processus d'importation échoue. L'ordre des colonnes dans le fichier CSV doit suivre le format d'un fichier de modèle ou d'un fichier CSV exporté, et ne peut pas être changé.

/path <file_path>

Indique le chemin d'accès et le nom du fichier d'importation.

/key <key>

Indique la clé d'administrateur de CA Business Service Insight unique.

/secret <secret>

Indique le mot de passe d'administrateur de CA Business Service Insight unique.

Exemple : importation des données de service à partir d'un fichier CSV local

L'exemple ci-dessous illustre la commande d'importation des données de service à partir d'un fichier CSV local avec utilisation de HTTPS.

```
InsightUtil /importservice /https /server name001 /port 8443
/format csv /path C:\MyFiles\file.csv /key mykey /secret
mypassword
```

Exportation d'un fichier de modèle de service

Si vous envisagez d'importer manuellement des données de service dans CA Business Service Insight à l'aide de la fonctionnalité InsightUtil, votre fichier CSV ou XML doit être au bon format. Vous pouvez exporter un fichier de modèle de service à partir de CA Business Service Insight au format CSV ou XML, puis ajouter vos données de service avant de l'importer.

Procédez comme suit :

Exécutez la commande suivante :

```
InsightUtil /exporttemplate /format csv|xml /path <file_path>
```

Le fichier de modèle exporté contient toutes les informations de mise en forme nécessaires et quelques exemples de données.

Annexe A : Exemples de domaines de service et de catégories de domaine

Le tableau suivant est une compilation des domaines de service et des catégories de domaine courants communément utilisés.

Domaine de service	Catégorie de domaine	Commentaires
Disponibilité du système	% de temps disponible	
	Nombre d'interruptions/temps d'indisponibilité	
	Durée moyenne de réparation	
	Durée moyenne entre deux pannes	
Disponibilité du service	Minutes d'indisponibilité	
	Nombre de jours perturbés	
Gestion financière	Coût du service	
Performance du processus	% de processus terminés en temps et en heure	
	Nombre de cycles de processus terminés	
Gestion des incidents	% d'incidents résolus < T1	
	% d'incidents auxquels ayant obtenu une réponse < T1	
	Nombre d'incidents	
	% d'incidents transformés en problèmes	
Satisfaction du client	% de satisfaction du client	
	Score CSI moyen	
Performance du centre d'assistance	% d'appels abandonnés	
	Délai de réponse à l'appel moyen	
	% TRPN	(taux de résolution de premier niveau)

Domaine de service	Catégorie de domaine	Commentaires
Qualité des données	% de précision	
	% d'opportunité	
	Nombre d'erreurs/défaillances	

Annexe B : Exemples d'étude de cas

Ce chapitre traite des sujets suivants :

Exemples de modélisation de contrat (page 271)
Exemple de modélisation de métrique financière (page 284)
Exemples de modélisation de données (page 292)
Exemple d'utilisation d'attributs personnalisés (page 304)
Exemples de script de conversion (page 308)
Exemples d'écriture de scripts de logique applicative (page 314)
Ecriture d'exemples de logique applicative efficaces (page 331)
Etude de cas 19 : Assistant de configuration d'adaptateur pour une source de données basée sur un fichier (page 349)
Etude de cas 21 : Exemple d'intégration de LDAP (page 367)
Etude de cas 22 : Génération de rapports à l'aide de PERL (page 374)

Exemples de modélisation de contrat

Pour chacune des études de cas suivantes, utilisez les éléments de catégorisation suivants pour les objectifs décrits :

- Service
- Domaine de service
- Catégorie de domaine
- Période d'application
- Cible
- Période de suivi
- Unité de mesure
- Descriptif de calcul
- Paramètres de contrat\métrique

Etude de cas 1 : Disponibilité du système

Examinez la garantie contractuelle suivante :

"Le système X doit être disponible pour 99,6 % du mois au minimum pendant des heures d'activité professionnelle."

Cette situation peut être décrite en utilisant les entités CA Business Service Insight suivantes :

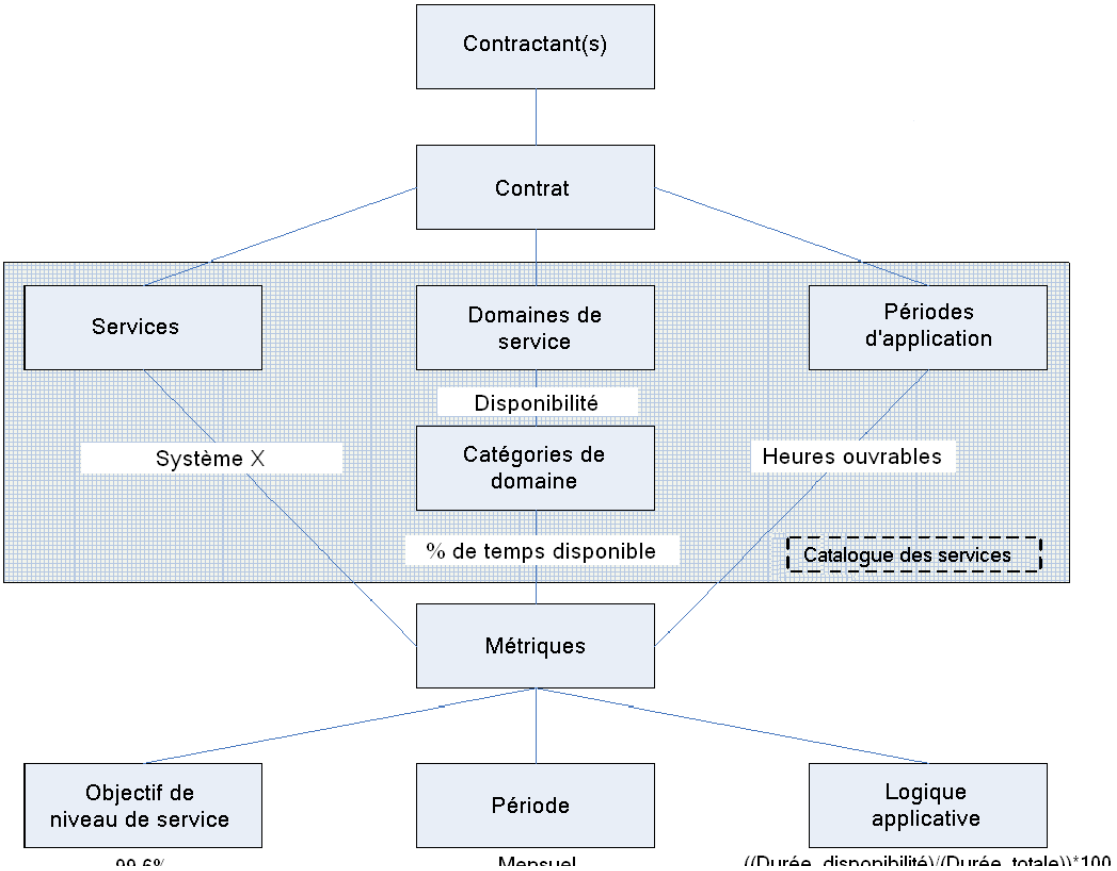
- **Nom de la métrique** : % de temps disponible du système X
- **Période de suivi :** 1 mois
- **Période d'application** : Heures d'activité professionnelle (doivent être davantage définies ultérieurement)
- **Logique applicative** : ((temps disponible)/(temps total))*100

  Remarque : Cette étude de cas n'est concernée que si le suivi se déroule pendant des heures d'activité professionnelle (qui sont la période d'application de la métrique)
- **Cible** : 99,6 %

Outre les informations précédentes sur les métriques, il est également possible de déduire les éléments suivants du catalogue de services système à partir de la description susmentionnée :

- **Service** : Système X.
- **Domaine de service** : Disponibilité.
- **Catégorie de domaine** : % du temps disponible.
- **Groupe de services** : Un groupe qui identifie plus d'un système pour lequel un suivi peut être assuré. A cette étape, il est difficile d'établir s'il est possible de créer un groupe convenable.

A présent, vous pouvez donc reproduire le diagramme de la section Modélisation de contrat de ce document qui affiche ces entités dans un diagramme.



Etude de cas 2 : Disponibilité du système 2

La disponibilité du Système CNP doit maintenir les niveaux suivants :

Environnement	Jours ouvrables	Week-ends
Production	99,9 %	98,9 %
Développement	90 %	N/D
Essai/QA	Aucune garantie	N/D
Réseau	99,9 %	N/D

Solutions Suggérées :

Métrique	Moyenne de production du système CNP pendant les jours ouvrables
Cible	99,9
Période de suivi	1 mois

Unité de mesure	%
Service	Production du système CNP
Domaine de service	Disponibilité
Catégorie de domaine	Disponibilité de l'application
Période d'application	jours ouvrables
Métrique	**Moyenne de production du système CNP pendant les week-ends**
Cible	98,9
Période de suivi	1 mois
Unité de mesure	%
Service	Production du système CNP
Domaine de service	Disponibilité
Catégorie de domaine	Disponibilité de l'application
Période d'application	Week-ends
Métrique	**Moyenne de développement du système CNP pendant les jours ouvrables**
Cible	90
Période de suivi	1 mois
Unité de mesure	%
Service	Développement du système CNP
Domaine de service	Disponibilité
Catégorie de domaine	Disponibilité de l'application
Période d'application	jours ouvrables
Métrique	**Moyenne d'essai/QA du système CNP pendant les jours ouvrables**
Cible	aucun
Période de suivi	1 mois
Unité de mesure	%
Service	Q/A d'essai du système CNP
Domaine de service	Disponibilité

Catégorie de domaine	Disponibilité de l'application
Période d'application	jours ouvrables
Métrique	**Disponibilité réseau du système CNP**
Cible	99,9
Période de suivi	1 mois
Unité de mesure	%
Service	Réseau du système CNP
Domaine de service	Disponibilité
Catégorie de domaine	Disponibilité réseau
Période d'application	Toujours

Etude de cas 3 : Temps de réponse du système

L'étude de cas suivante illustre des métriques de temps de réponse. Un contrat peut être modelé de plusieurs façons, chacune ayant ses avantages.

L'exemple suivant examine diverses méthodes de modélisation :

Modélisation suggérée, solution A	**Valeur maximum**
Temps de réponse de la transaction du système CNP	Ne peut pas dépasser 750 millisecondes par mois
Nom de la métrique	Temps de réponse maximum de la transaction
Cible	750
Période de suivi	1 mois
Unité de mesure	Millisecondes
Période d'application	Toujours
Service	Système CNP
Domaine de service	Performances
Catégorie de domaine	Temps de réponse maximum de la transaction

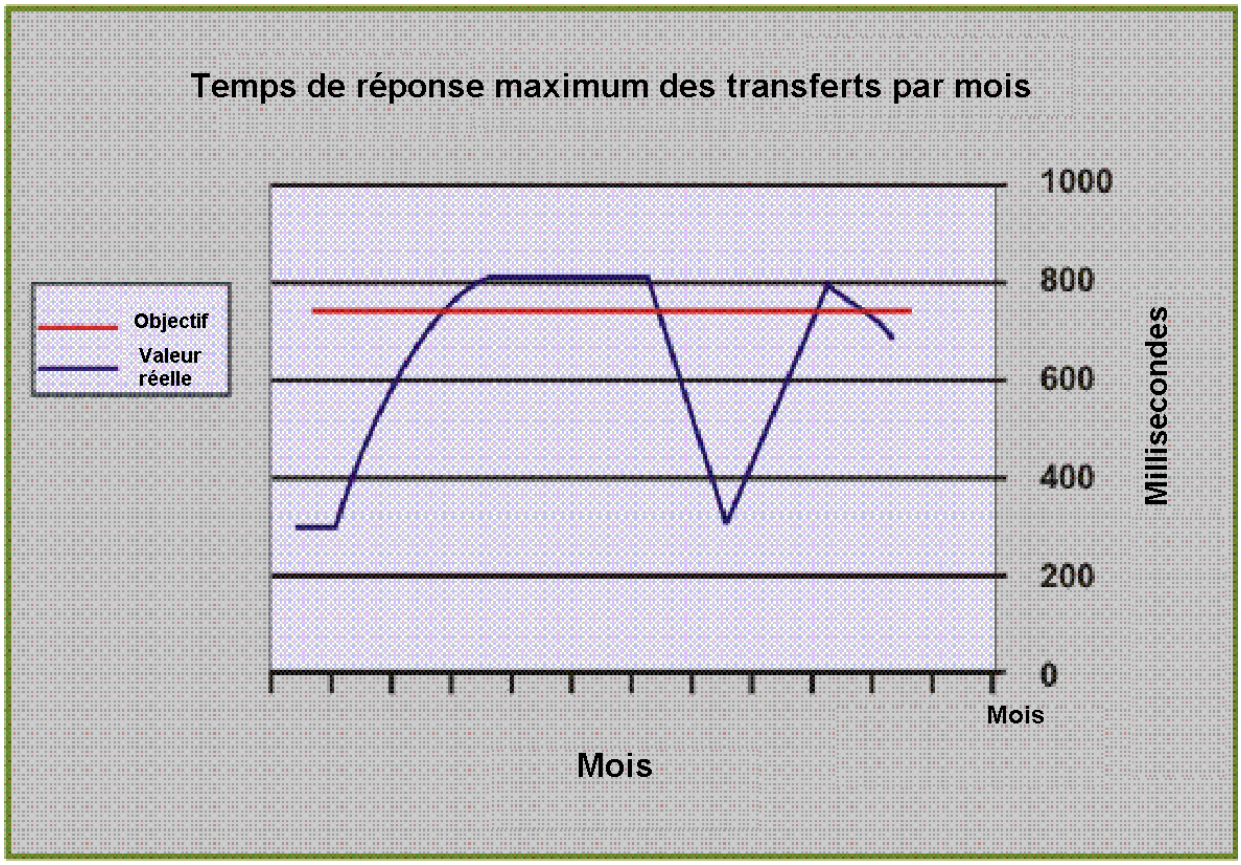
D'après la matrice susmentionnée, comment le niveau de service réel est-il calculé ?

D'après la définition de la catégorie de domaine, il semble que le niveau de service réel soit calculé comme valeur maximale. Cela implique que, parmi toutes les transactions accomplies pendant un mois, la transaction ayant la valeur maximale est capturée et cette valeur est comparée à la cible.

Remarque : Le calcul du niveau de service se fonde sur une agrégation de données brutes pendant une période donnée. Pendant chaque période, la métrique fournit un résultat unique. La cible d'une métrique n'est pas comparée à une transaction unique, mais à un résultat mensuel qui est une agrégation périodique de toutes les transactions durant cette période. Le gestionnaire de contrats doit veiller à ce que ce résultat reflète, d'une part, le contrat, et, d'autre part, la qualité du service.

Notez que la mesure du temps de réponse comme valeur maximale est une obligation très stricte et très difficile à accomplir en pratique. Mesurer un niveau maximal signifie qu'une transaction unique de 751 ms sur un million de transactions réalisées au cours d'un mois suffit pour causer une infraction du contrat. Toutes les barres dans les rapports seront par conséquent rouges et ne refléteront pas la qualité réelle du service qui a été fourni.

L'illustration suivante décrit un rapport typique dans ces circonstances.


Temps de réponse maximum des transferts par mois
Objectif
Valeur réelle
1000
800
600
400
200
0
Millisecondes
Mois
Mois

Une transaction qui dépasse la cible sera considérée comme violation du contrat. C'est une très mauvaise base pour estimer la qualité réelle du service fourni, car elle reflète uniquement une transaction unique et rien n'apparaît quant au reste des transactions. Par exemple, s'agit-il d'une défaillance unique ou d'une tendance ? S'il ne s'agit pas d'un incident isolé, combien de défaillances ont-elles été constatées ou quelle est la proportion de transactions échouées par rapport au nombre total de transactions accomplies au cours du mois ? Il peut y avoir un certains nombre de mois présentant de telles occurrences, qui constituent ainsi une violation du contrat, mais quelle est la tendance ? S'améliore-t-elle ou s'aggrave-t-elle ? Voilà autant de questions que le **gestionnaire de niveau de service** peut poser et le rapport doit pouvoir lui fournir des réponses.

Remarque : En définissant la métrique et le descriptif de calcul s'y rapportant, il est très important d'envisager comment les résultats s'afficheront dans un rapport. Ce rapport doit fournir deux éléments cruciaux :

- Y-a-t-il eu infraction au contrat ?
- Le rapport doit offrir aux gestionnaires suffisamment de transparence au niveau des données et de capacité pour examiner la cause première de la défaillance et leur apporter les outils nécessaires pour comprendre pleinement leurs composants de service.

Modélisation suggérée, solution B	**Temps moyen de réponse**
Temps de réponse de la transaction du système CNP	Ne doit pas excéder 750 millisecondes par mois
Métrique	Temps de réponse moyen de la transaction
Cible	750
Période de suivi	1 mois
Unité de mesure	Millisecondes
Catégorie de domaine	Temps de réponse moyen de la transaction

Le calcul du temps de réponse moyen donne une meilleure idée de la qualité mensuelle de service et peut à la fois encore refléter ces mois avec des temps de réponse extrêmes ou hors-contrat.

Solution C de modélisation suggérée :	**Pourcentage de transactions qui ont été correctement accomplies en dessous du seuil.**
Temps de réponse de la transaction du système CNP	Ne doit pas excéder 750 millisecondes par mois
Métrique	Temps de réponse pour la transaction réussie
Cible	100
Période de suivi	1 mois
Unité de mesure	% de réussite
Paramètre de métrique	750 ms
Service	Système CNP
Domaine de service	Performances
Catégorie de domaine	Temps de réponse pour la transaction réussie
Période d'application	Toujours

Dans le cadre de cette méthode, le calcul détermine le pourcentage de transactions qui ont été correctement accomplies sous le seuil des 750 ms pendant la période spécifiée, donnée par la formule :

*((Nombre de transactions en dessous de 750 ms.)/Nombre total de transactions))*100*

L'expression de la garantie en taux de réussite fournit la capacité de conserver une garantie stricte (cible 100 %), tout en tenant également compte de la valeur réelle représentant la qualité du service.

Avec cette méthode, la cible n'est désormais plus une limite supérieure de 750 ms, mais la proportion à maintenir. Dans les cas où la garantie doit être stricte, la cible doit être de 100 %, ce qui ne laisse aucune place à la défaillance. Remarquez que, dans ce cas, une variable supplémentaire est introduite, à savoir le paramètre de la métrique. Ce paramètre doit être mis en oeuvre comme paramètre de métrique pour activer des modifications simples, le cas échéant.

L'imposition d'un modèle de type progressif peut présenter un modèle de type escalade :

Les solutions suivantes définissent trois métriques au lieu d'une métrique unique, comme dans les solutions précédentes.

Métrique	Temps de réponse pour la transaction réussie
Cible	95
Période de suivi	1 mois
Unité de mesure	% de réussite
Paramètre de métrique	750 ms
Métrique	Temps de réponse pour la transaction réussie
Cible	99
Période de suivi	1 mois
Unité de mesure	% de réussite
Paramètre de métrique	850 ms
Métrique	Temps de réponse pour la transaction réussie

Cible	100
Période de suivi	1 mois
Unité de mesure	% de réussite
Paramètre de métrique	1000 ms

S'il est nécessaire de signaler l'obligation contractuelle, ainsi que le nombre de transactions dépassant le seuil des 750 ms, vous devez définir une métrique supplémentaire pour calculer le nombre de transactions qui ont échoué.

Remarque : Chaque métrique génère un résultat unique pendant une période donnée. Si elle est définie pour calculer le pourcentage de transactions, elle ne peut pas également fournir le nombre de transactions.

La seule façon de produire des rapports supplémentaires à partir d'une métrique est d'utiliser les sorties de la logique applicative. (Consultez la section Sorties - Tables utilisateur (page 178) qui traite de la sortie des résultats de la logique applicative).

Métrique	**Nombre de transactions échouées**
Cible	Aucune cible
Période de suivi	1 mois
Unité de mesure	Nombre de transactions
Paramètre de métrique	750 ms
Service	Système CNP
Domaine de service	Performances
Catégorie de domaine	Nombre de transactions
Période d'application	Toujours

Etude de cas 4 : Performance du centre d'assistance

Etude de cas illustrant une situation de centre d'assistance

Le centre d'assistance doit afficher une réussite à 100 % en accomplissant tout ce qui suit :

Type de ticket	**Durée de résolution**
Priorité 1	1 heure

Priorité 2	2 heures
Priorité 3	4 heures

Modélisation suggérée, solution A :

Métrique	**Temps de résolution pour la priorité 1**
Cible	100
Période de suivi	1 mois
Unité de mesure	% de réussite
Paramètre de contrat	Matrice d'heure de résolution
Service	Centre d'assistance
Domaine de service	Performance du centre d'assistance
Catégorie de domaine	Temps de résolution pour le ticket
Période d'application	Toujours

La matrice susmentionnée s'applique à trois métriques. Pour chaque priorité, une métrique à part est définie avec toutes les priorités dans les mêmes catégories.

Modélisation suggérée, solution B :

La définition de la métrique reste la même comme illustré en solution A.

Option 1 :

Service	Centre d'assistance
Domaine de service	Priorité de gestion de ticket 3
Catégorie de domaine	Temps de résolution pour le ticket
Catégorie de domaine	Temps de réponse du ticket
Période d'application	Toujours

Option 2 :

Service	Centre d'assistance
Domaine de service	Gestion de ticket
Catégorie de domaine	Priorité 3 temps de résolution du ticket
Catégorie de domaine	Temps de réponse du ticket

Période d'application	Toujours

Etude de cas 5 : Sauvegarde de système

La sauvegarde s'accomplit comme suit :

Délai	**Nombre de sauvegardes**
hebdomadaire	6
mensuel	27
annuel	350

Solutions suggérées :

Métrique	**Nombre de sauvegardes hebdomadaires**
Cible	6
Période de suivi	1 semaine
Unité de mesure	Sauvegardes
Service	Sauvegarde
Domaine de service	Sauvegarde
Catégorie de domaine	Nombre de sauvegardes par semaine
Période d'application	Toujours
Métrique	**Nombre de Sauvegardes Mensuelles**
Cible	27
Période de suivi	1 mois
Unité de mesure	Sauvegardes
Service	Sauvegarde
Domaine de service	Sauvegarde
Catégorie de domaine	Nombre de sauvegardes par semaine
Période d'application	Toujours

Métrique	Nombre de sauvegardes annuelles
Cible	350
Période de suivi	1 an
Unité de mesure	Sauvegardes
Service	Sauvegarde
Domaine de service	Sauvegarde
Catégorie de domaine	Nombre de sauvegardes par semaine
Période d'application	Toujours

Exemple de modélisation de métrique financière

L'étude de cas suivante présente un exemple de modélisation financière.

Etude de cas 6 : Modélisation de conditions financières d'un contrat/service

Il existe trois types généraux de métriques utilisés pour la modélisation des conditions financières d'un service ou contrat, à savoir :

- Coûts de prix fixe
- Coûts de consommation
- Charges de pénalité/prime

Considérez l'exemple suivant :

Une nouvelle société émerge et requiert un service de courriel accompagné de sa configuration et maintenance de ses boîtes aux lettres. Le nombre de boîtes aux lettres augmentera évidemment lorsque de nouveaux employés seront engagés. L'approvisionnement d'un service de courriel pour un contrat est fixé à un prix de 1000 $, plus un coût supplémentaire mensuel par boîte aux lettres. Ce coût par boîte aux lettres est échelonné et tarifé comme suit :

Nombre de boîtes aux lettres	Coût par boîte aux lettres
1-1000	1,00 $
1001 - 5000	0,80 $

5001+	0,50 $

De fait, plus le nombre de boîtes aux lettres ajoutées est élevé, plus le coût supplémentaire est faible. (Par exemple : 1500 boîtes aux lettres coûteront (1000 x 1 $) + (500 x 0,80 $) = 1400 $.) L'utilisation de ce modèle vous permet de produire deux métriques qui refléteront ceci dans le contrat.

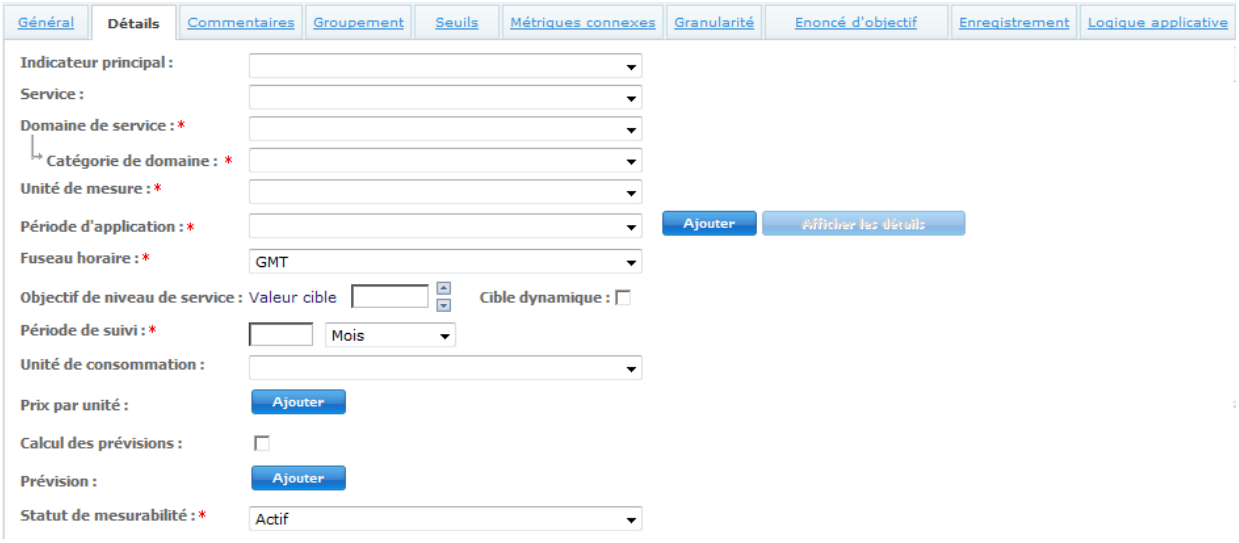
- Coût de service de courriel (fixe)
- Coût d'utilisation de la boîte aux lettres (selon la quantité/consommation)

Une estimation du nombre de membres du personnel est également effectuée par l'équipe de gestion durant l'année (2007), établie comme suit. La tendance est due à la croissance initiale de la société de par ses emplois, puis des nouveaux bureaux ouverts dans d'autres régions :

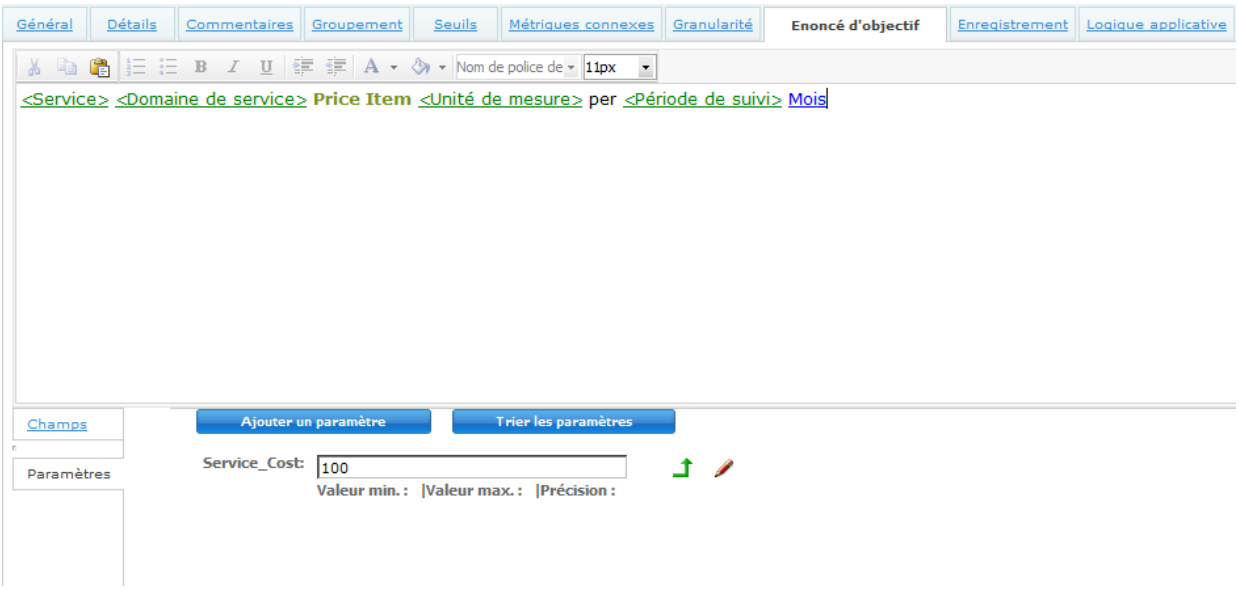
Jan	Fév	Mars	Avr	Mai	Juin	Juil	Août	Sept	Oct	Nov	Déc
50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

Pour modeler ces métriques, procédez comme suit :

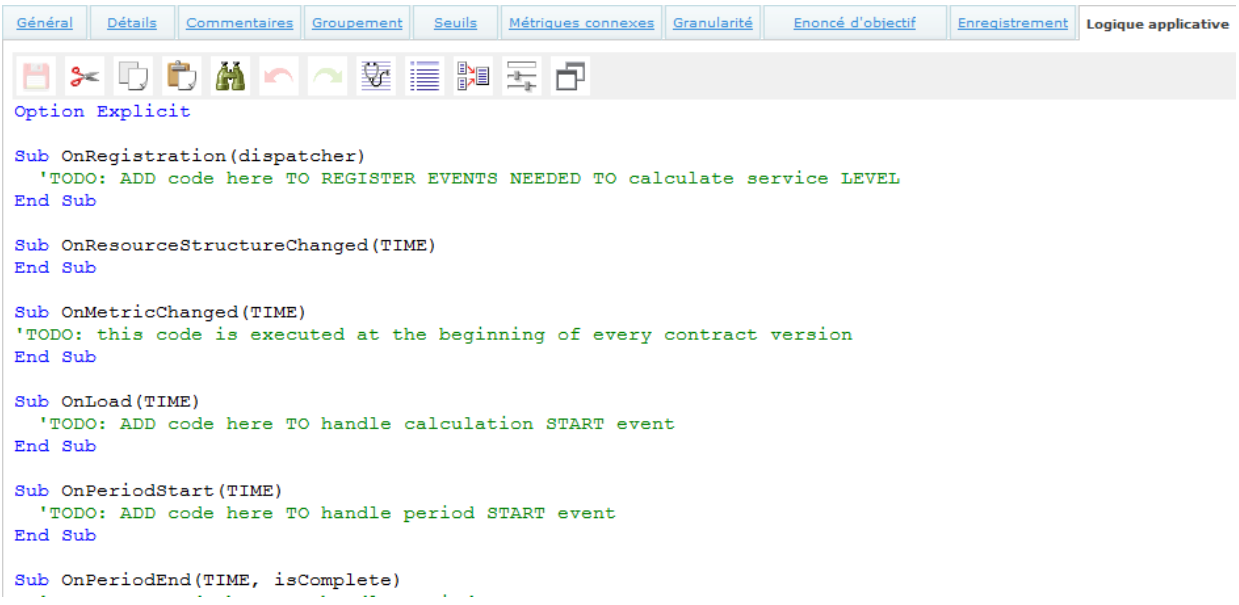
Créez une métrique de coût fixe à l'aide du type d'élément de prix dans le contrat et spécifiez les détails suivants :



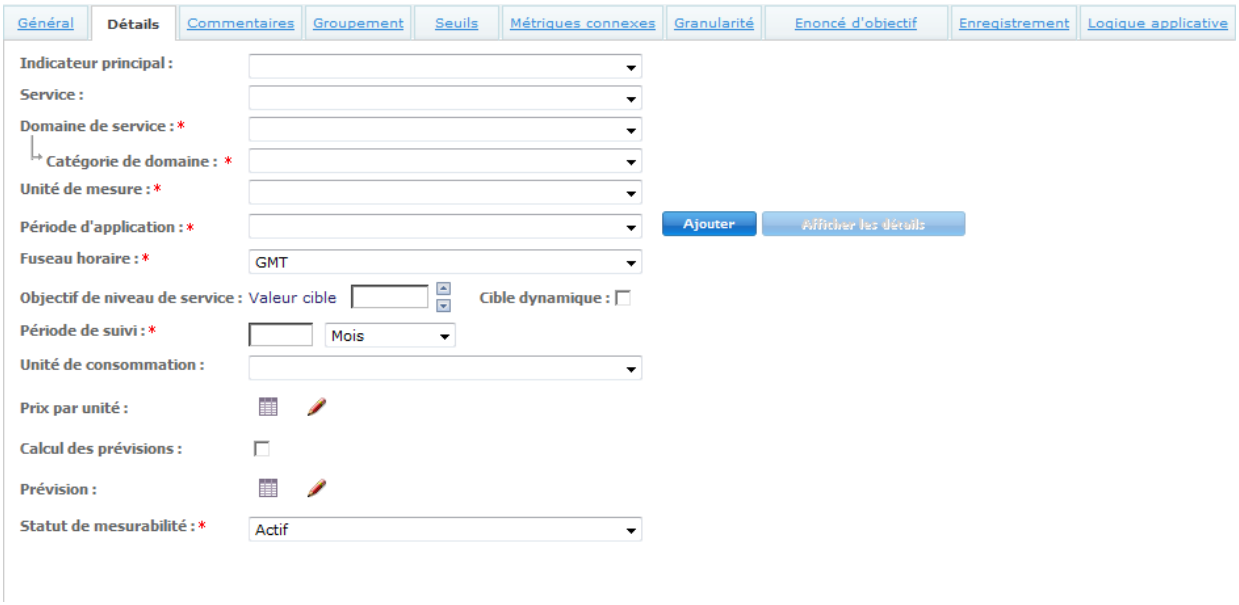
Pour spécifier le coût fixe dans le contrat de ce service, implémentez-le comme paramètre dans la logique applicative (où le coût fixe doit être renvoyé depuis la fonction Résultat). Ce paramètre peut alors être accessible via l'énoncé d'objectif de la métrique, tel qu'affiché ci-dessous :



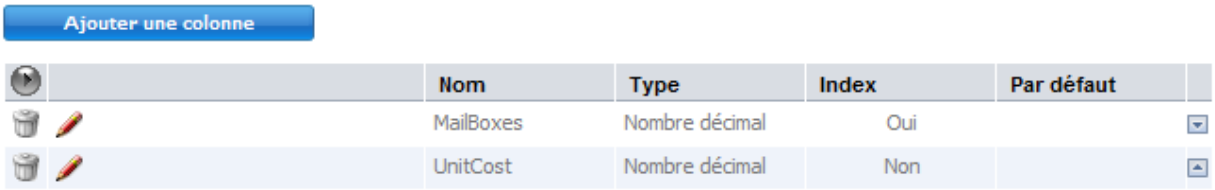
Le renvoi de la valeur de paramètre pour cette métrique consiste simplement à renvoyer la valeur du coût de service via la fonction Résultat.



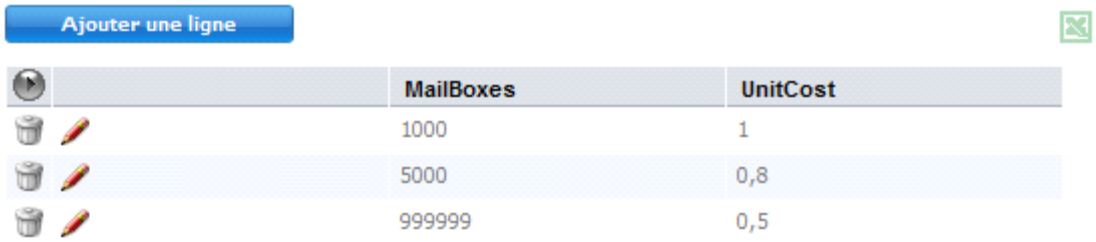
Créez ensuite la métrique de tarification variable (à nouveau, utilisez le type d'élément de prix) pour déterminer les coûts de consommation du nombre de boîtes aux lettres utilisées. Nommez cette métrique *Coût de consommation de la boîte aux lettres* et créez-la grâce aux détails suivants :



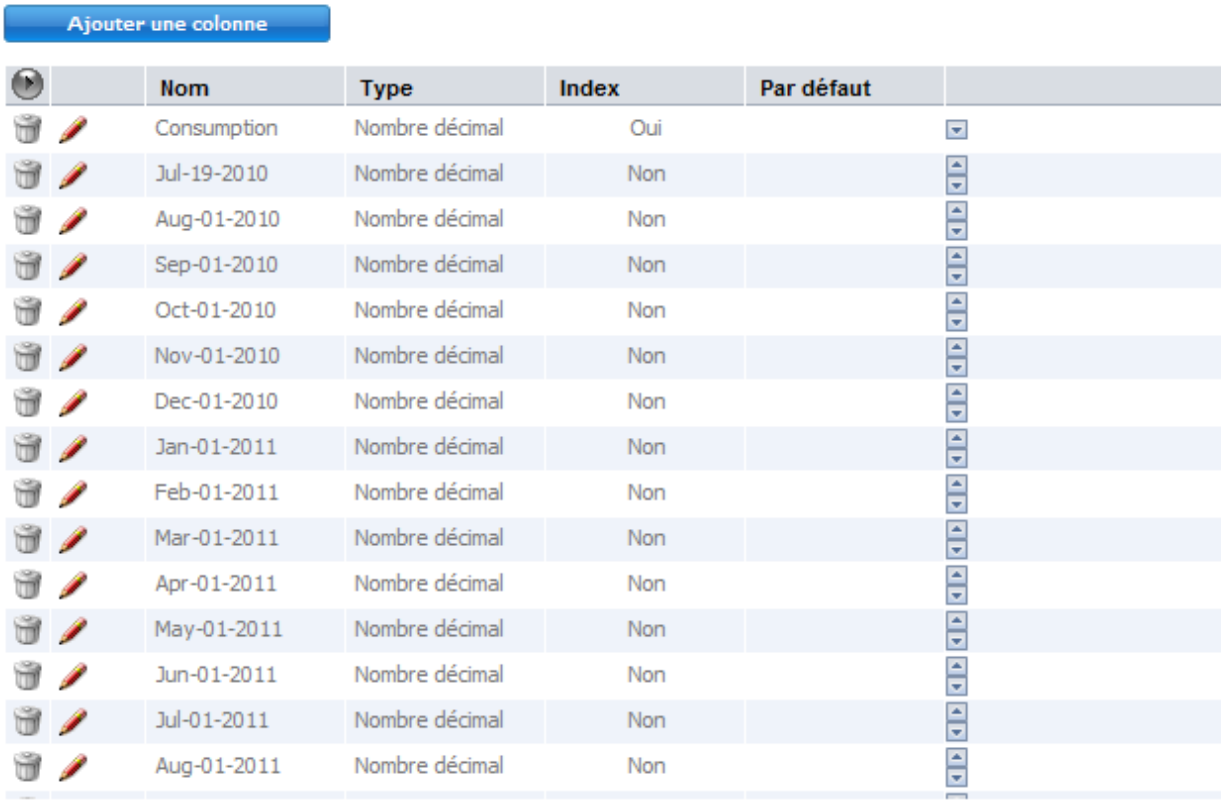
Dans cette instance, vous devez entrer les paramètres de consommation dans les détails de la métrique. Ceux-ci iront dans la table Prix par unité. Pour modeler la table ci-dessus sur le nombre de boîtes aux lettres selon le coût, créez une colonne pour la limite supérieure de boîtes aux lettres et une autre pour les prix unitaires :



Entrez alors les valeurs pour chaque niveau. Dans ce cas, la limite supérieure de boîtes aux lettres détermine le niveau de coût associé. Etant donné qu'il existe 3 niveaux, ceux-ci sont ajoutés à la table de la manière suivante :



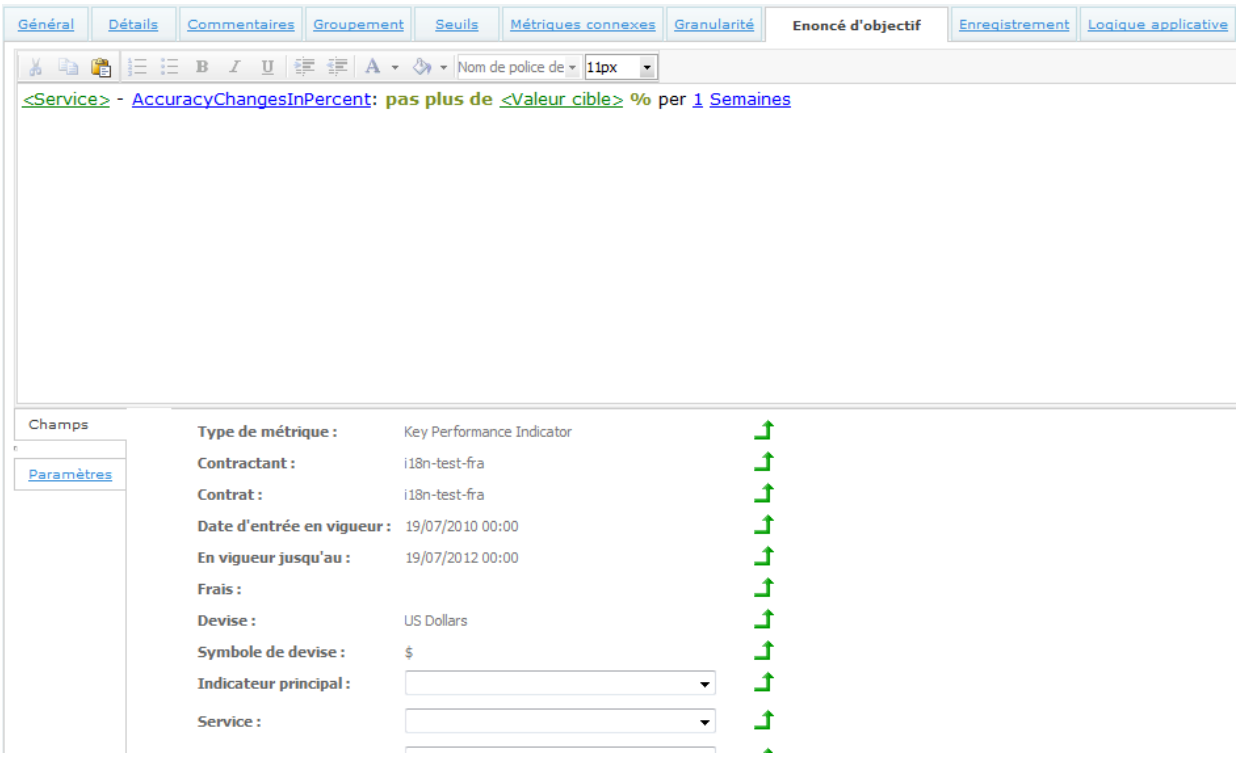
Ajoutez à cela l'implémentation de la fonction de prévision à la consommation des boîtes aux lettres. Procédez en créant la table de prévision avec le schéma Mensuel préétabli.



La table est alors remplie avec les valeurs des tables données dans la description de scénario.

Consommation	01-Janv. 2007	01-Fév. 2007	01-Mars 2007	01-Avr 2007	01-Mai 2007	01-Juin 2007	01-Juil 2007	01-Août 2007	01-Sept 2007	01-Oct 2007	01-Nov 2007	01-Déc 2007
1	50	100	500	900	1600	1700	1800	2500	2600	3500	3600	5800

Vous pouvez maintenant ajouter l'énoncé d'objectif de la métrique. Dans ce cas, aucune valeur de paramètre n'est requise car celles-ci sont dérivées des tables Prix par unité et Prévision.



Finalement, terminez la logique applicative de la façon suivante :

```
Option Explicit

Dim PPUmap1, PPUmap2, PPUmap3, PPUkey, FCmap, periodFC, TierPPU
Dim currentMonth, TotalMailboxes, MailboxesThisPeriod, TotalPrice

Sub OnRegistration(dispatcher)
   'exemple d'enregistrement seulement
   dispatcher.RegisterByMetric "OnMailboxAddedEvent", "NewMailboxEventType", _
      "MailboxResource", "MONTH", "MetricName", "MetricContract", _
      "MetricContractParty"
End sub

Sub OnLoad(TIME)
   'Initialise les plans de tarification et les plans de prévision
   Set PPUmap1 = Context.Field ("Price Per Unit")(1)
   Set PPUmap2 = Context.Field ("Price Per Unit")(2)
   Set PPUmap3 = Context.Field ("Price Per Unit")(3)
   Set FCmap = Context.Field ("Forecast")(1)
End sub

Sub OnPeriodStart(TIME)
   'A FAIRE : AJOUTEZ du code ici POUR gérer l'événement de DEMARRAGE de la
période
   currentMonth = GetMonth (time)
   If Context.IsInForecast Then
      periodFC = getForecastValue (currentMonth)
   End If
   MailboxesThisPeriod = 0
   TotalPrice = 0
End sub

Sub OnPeriodEnd(TIME, isComplete)
   ' Calculer le prix actuel de toutes les boîtes aux lettres en utilisant le
modèle
   ' de tarification échelonné
   ' Ceci utilise une approche cumulative en traversant chaque niveau pour
   ' déterminer le coût total.
   TotalPrice = getMailboxCost (TotalMailboxes)
End sub

Sub OnTimeslotEnter(TIME)
End sub

Sub OnTimeslotExit(TIME)
End sub

Sub OnMailboxAddedEvent(eventDetails)
   MailboxesThisPeriod = MailboxesThisPeriod + 1
```

```
   TotalMailboxes = TotalMailBoxes + 1
End sub

Function Forecast
   Forecast = getMailboxCost (periodFC)
End Function

Function Target
   Target = Null
End Function

Function Result
   result = TotalPrice
End Function

Function getforecastvalue(q)
   getforecastvalue = FCmap (q)
End Function

Function getmonth(time)
   'cette fonction récupère le mois
   Dim lTime
   lTime = Tools.GetLocaleTime(time)
   getmonth = monthname (datepart ("m", lTime), True) & _
      "-0" & datepart ("d", lTime) & "-" & datepart ("yyyy", lTime)
End Function

Function getMailboxCost(num_boxes)
   ' Fonction calculant le coût des boîtes aux lettres en utilisant le modèle de
tarification échelonné
   Dim returnValue
   If num_boxes <= PPUmap1 ("Mailboxes") Then
      ' Premier niveau
      returnValue = num_boxes * PPUmap1 ("UnitCost")
      'Out.Log "Tier1: " & num_boxes
   Else If num_boxes > PPUmap1 ("Mailboxes") And num_boxes <= PPUmap2
("Mailboxes") Then
      ' Deuxième niveau uniquement
      returnValue = (PPUmap1 ("Mailboxes") * PPUmap1 ("UnitCost")) + _
         ((num_boxes - PPUmap1 ("Mailboxes")) * PPUmap2 ("UnitCost"))
      'Out.Log "Tier2: " & num_boxes
   Else If num_boxes > PPUmap2 ("Mailboxes") Then
      ' Troisième niveau
      returnValue = (PPUmap1 ("Mailboxes") * PPUmap1 ("UnitCost")) + _
         ((PPUmap2 ("Mailboxes") - PPUmap1 ("Mailboxes")) * PPUmap2 ("UnitCost"))
+ _
         ((num_boxes - PPUmap2 ("Mailboxes")) * PPUmap3 ("UnitCost"))
      'Out.Log "Tier3: " & num_boxes
   End If
```

```
   getMailboxCost = returnValue
   'Out.Log "Cost is: " & returnValue
End Function
```

Remarque : Ce script de logique applicative traite aussi bien le calcul de prévision (par l'utilisation de la table Prévision) que les résultats de coût de consommation financiers. Les deux utilisent la même formule getMailboxCost() calculant la tarification échelonnée selon la table Prix par unité définie pour cette métrique.

Exemples de modélisation de données

Les deux cas suivants illustrent le processus basique de modélisation de données ainsi que quelques points plus détaillés pouvant être impliqués.

Parce que le processus de modélisation de données est exécuté après le processus de modélisation de contrat, les calculs requis dérivent des garanties du contrat déjà connues, identifiées et définies comme partie du processus de modélisation de contrat.

Le modèle de données doit tenir compte de tous les calculs nécessaires.

Dans les deux études de cas suivantes, une ou plusieurs conditions requises sont choisies pour démontrer le processus de modélisation de données.

Etude de cas 7 : Performance de serveur

Voici une étude de cas typique sur la performance de serveur.

Considérant la structure de source de données suivante :

Indication	Serveur	Mesure	Horodatage
disponibilité	Appserv01	1	03/01/2001 07:44
temps de réponse	Appserv01	354,6	01/03/2001 09:58
Charge d'UC	Dbserv02	83 %	01/03/2001 12:12
disponibilité	Appserv01	0	01/03/2001 14:26
Charge d'UC	Dbserv02	94,30 %	01/03/2001 16:40
capacité	Firewall01	10 %	01/03/2001 18:54
temps de réponse	Dbserv02	476,89	01/03/2001 21:08

Indication	Serveur	Mesure	Horodatage
disponibilité	Appserv02	1	01/03/2001 21:24
temps de réponse	Appserv01	774,4	01/03/2001 21:48
Charge d'UC	Dbserv01	83 %	01/03/2001 21:52

S'ajoutent aux conditions requises ci-dessus les calculs nécessaires suivants :

Calcul du pourcentage de disponibilité de chaque serveur d'applications.

La disponibilité de chaque serveur doit être calculée séparément. De fait, le calcul de disponibilité d'un serveur unique nécessite la réception d'événement pour ce serveur seulement. De plus, les sources de données contiennent d'autres indicateurs de performance non pertinents dans les calculs de disponibilité (temps de réponse, capacité et ainsi de suite), les indicateurs de disponibilité et le serveur concerné doivent donc être filtrés.

Les critères de filtre dans CA Business Service Insight étant Type d'événement et Ressources, convertissez les critères de filtre depuis les valeurs de source de données en une définition de Type d'évenement et Ressource.

Dans ce cas, l'indicateur est une valeur idéale pour la configuration en type d'événement dans CA Business Service Insight, car il décrit logiquement le type d'événement. Il n'existe qu'un nombre limité de types : la disponibilité, le temps de réponse, la capacité et la charge d'UC. Cela signifie que pour la métrique qui calcule la disponibilité pour le serveur, l'enregistrement concerne le type d'événement de disponibilité.

Dans ce cas, lorsqu'un trop grand nombre de serveurs est présent et qu'il est nécessaire de calculer la disponibilité de chacun, chaque serveur doit être défini en tant que Ressource. Ils doivent alors être groupés au sein d'un groupe de ressources, et la métrique sera groupée sur celui-ci.

Modélisation suggérée :

Nom d'événement Evénement de disponibilité.

Comportement Signalé comme changement de statut en 0 ou 1.

Champ Horodatage Horodatage (unique champ de date dans la source de données).

Champ Ressource Serveur (tous les serveurs qui s'affichent dans la source de données sont convertis en une ressource CA Business Service Insight).

Champ Type d'événement	Indicateur (chacune des valeurs de ce champ est convertie en un type d'événement dans CA Business Service Insight. Il existe quatre types d'événement).
Champs de données	La mesure est 0 ou 1 (uniquement pour les enregistrements de disponibilité).

Les allocations de ressource suivantes doivent être définies :

Attribut Type de ressource	Serveur d'applications
Allocation au contractant	Chaque serveur d'applications est alloué au contractant, où le serveur concerné exécute son application. Cela a pour effet d'activer l'enregistrement par le contractant, qui récupère en conséquence tous les serveurs.
Allocation au service	Voir ci-dessus.
Allocation au groupe de ressources	Facultatif. Il est souvent nécessaire de grouper les ressources lorsqu'un regroupement est requis.

En dernier lieu, basé sur toutes les définitions susmentionnées ;

Enregistrement par	Pour la métrique groupée, calculer la disponibilité de chaque serveur individuellement requiert l'enregistrement en ressource.

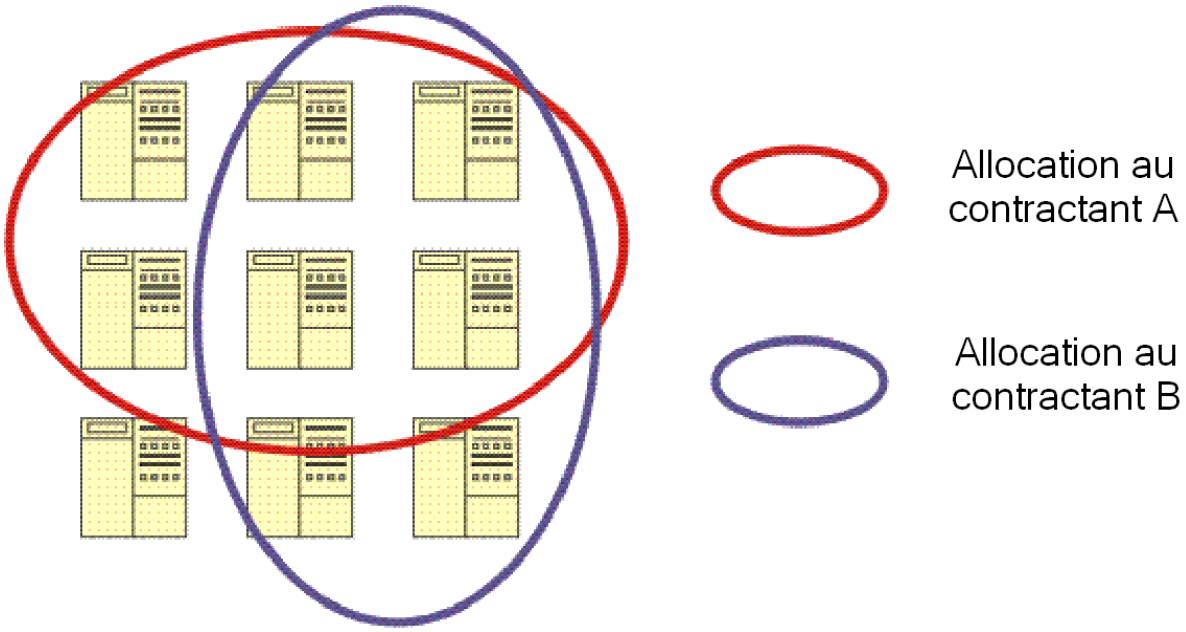
Afin de pouvoir remplir la condition susmentionnée, le critère suivant est ajouté :

Calculez de façon séparée le temps de réponse moyen des serveurs d'applications pour chaque contractant.

Pour cette condition, il est nécessaire de recevoir des événements de temps de réponse pour tous les serveurs d'applications faisant partie du groupe d'applications exécutant des applications pour le contractant concerné. Vous pouvez recevoir les événements de temps de réponse par l'enregistrement du type d'événement, lequel a été créé à partir du champ d'indice comportant la valeur de temps de réponse. Cela assure de ne recevoir que les événements qui rapportent les temps de réponses des serveurs.

Pour recevoir les événements qui ne concernent que des serveurs spécifiques à un contractant, les ressources doivent être enregistrées via l'allocation du contractant.

Vous pouvez allouer une ressource à plus d'un contractant unique, service ou groupe. Il peut donc arriver qu'un événement envoyé pour le calcul du temps de réponse faisant partie du contrat du contractant A fasse également partie des calculs du contractant B.



Etude de cas 8 : Performance du centre d'assistance

Voici un cas d'étude typique sur la performance du centre d'assistance.

Considérant la source de données affichée ci-dessous :

N° TK	Priorité TK	Date d'ouverture	Date de fermeture	Date de résolution	Ref pers	Emplacement
3800968	1	19/12/2003 07:51	05/01/2004 11:31	22/12/2003 12:00	CM3	Londres
38000509	1	18/12/2003 09:21	05/01/2004 11:33	22/12/2003 12:00	CM1	Ipswitch
38084199	2	07/01/2004 11:20	14/01/2004 09:10	09/01/2004 12:00	CM2	Ipswitch
38188329	3	21/01/2004 09:27	27/01/2004 09:09	24/01/2004 12:00	CM3	Leeds

N° TK	Priorité TK	Date d'ouverture	Date de fermeture	Date de résolution	Ref pers	Emplacement
37964069	3	12/12/2003 14:04	05/01/2004 11:35	19/12/2003 12:00	CM286	Birmingham
3796448	1	12/12/2003 14:18	05/01/2004 11:39	19/12/2003 12:00	CM263	Luton
37965039	2	12/12/2003 14:57	14/01/2004 15:05	18/12/2003 12:00	CM264	Leeds
37970699	2	15/12/2003 09:26	05/01/2004 11:22	22/12/2003 12:00	CM288	Londres
37997259	1	17/12/2003 15:58	05/01/2004 11:27	23/12/2003 12:00	CM302	Ipswitch
38000259	1	18/12/2003 09:11	06/01/2004 14:44	29/12/2003 12:00	CM340	Londres
38021049	1	22/12/2003 09:32	06/01/2004 14:28	31/12/2003 12:00	CM341	Londres

La source de données affichée ci-dessus liste en détails des tickets du centre s'assistance, tickets gérés pour chaque client aux différents emplacements où le client opère. Un emplacement unique peut également être partagé entre plusieurs clients.

Lors de l'utilisation de cette source de données, les trois conditions suivantes sont requises pour générer un rapport :

- % de tickets de priorité 1 résolus dans les quatre heures pour le client CM3.
- % de tickets de priorité 1 résolus dans les quatre heures pour le client CM3 à chaque emplacement.
- % de tickets de priorité 1 résolus dans les 24 heures pour le client CM3 à chaque emplacement.

Les conditions requises susmentionnées indiquent que les événements doivent être filtrés par :

- priorité
- client
- emplacement

Vous devez spécifier quel critère de filtration doit être converti en type d'événement et lequel en ressource appropriée.

Comment puis-je sélectionner un type d'événement ?

Des trois critères de filtration requis, la priorité du ticket est le plus à même à être converti en type d'événement, ce pour les raisons suivantes :

- Il décrit le type d'événement en cours de traitement (ticket de priorité 1).
- Le nombre de priorités existantes est relativement restreint (priorité 1, 2, 3).
- Le type d'événement lui-même est relativement constant (le centre d'assistance ne change que rarement la priorité selon laquelle le ticket est traité).

Comment puis-je sélectionner une ressource ?

Les deux autres critères de filtre requis, client et emplacement, sont les plus petites entités nécessitant un rapport. Par conséquent, la ressource est la combinaison du client et de l'emplacement.

Le client et l'emplacement sont des entités relativement fixes et qui ont un cycle de vie défini, de sorte que les nouveaux clients ou emplacements puissent être ajoutés. En outre, la relation entre un site et un client peut changer.

Il est également possible d'utiliser plus d'un champ depuis la source de données à des fins de conversion. Tandis que le champ de serveur des études de cas précédents était traduit en une ressource CA Business Service Insight, la ressource dans ce cas est la combinaison de deux champs. Chaque permutation produit par conséquent une nouvelle ressource.

La liste de ressources est affichée ci-dessous :

Champ client	Champ de site	Ressource de sortie
CM3	Londres	CM3@London
CM1	Ipswitch	CM1@Ipswitch
CM2	Ipswitch	CM2@Ipswitch
CM3	Leeds	CM3@Leeds
CM286	Birmingham	CM286@Birmingham
CM263	Luton	CM263@Luton
CM264	Leeds	CM264@Leeds
CM288	Londres	CM288@London
CM302	Ipswitch	CM302@Ipswitch
CM340	Londres	CM340@London

CM341	Londres	CM341@London

La ressource de sortie est une combinaison des champs client et emplacement qui utilise le symbole "+" pour les combiner. Il est important d'être conscient du nom de la ressource car il est extrait de la source de données et apparaît plus tard dans les rapports. Les capacités de génération de rapports doivent répondre aux attentes.

Remarque : Une des erreurs les plus courantes lors de la modélisation d'un centre assistance, ticket ou système d'incidents et de définir un seul incident comme ressource.

Les suppositions suivantes sont incorrectes et mènent souvent à des erreurs :

1. L'unique incident est celui qui est signalé. Ne définissez pas l'entité à signaler comme entité pour laquelle les calculs sont générés, de sorte que ce seul incident ne soit pas la base des calculs pour le site du client. En général, le SLA se base sur la performance globale et non la performance spécifique du traitement de ticket.
2. La garantie s'effectue par niveau de ticket. Il s'agit d'une erreur car les garanties sont périodiques, ce qui est calculé est une agrégation dans le temps.

Paramétrage des allocations pour les ressources

Pour la première condition de calcul :

1. % de tickets de priorité 1 résolus dans les quatre heures pour le client CM3.
   - Les événements de tickets attribuables à un client spécifique doivent être reçus. Le client fait partie de cette Ressource qui indique également le site du client. Il existe par conséquent deux options pour capturer tous les événements liés aux ressources et attribuables à ce client :
   - Dans les cas où le client de la source de données représente un contractant, les ressources peuvent être attachées au contractant approprié. Cela permet l'enregistrement par contractant. Lorsqu'elle est possible, cette solution est toujours préférable.
   - Créer un groupe de ressources pour chaque client de la source de données et y grouper toutes les Ressources appropriées. En utilisant cette méthode, l'enregistrement se fait par groupe de ressources.

   Concernant les deux conditions de calculs suivantes :
2. % de tickets de priorité 1 résolus dans les quatre heures pour le client CM3 à chaque emplacement.
3. % de tickets de priorité 1 résolus dans les 24 heures pour le client CM3 à chaque emplacement.

Vous devez, pour ces conditions, regrouper les Ressources en groupes de ressources. Les métriques nécessitent d'être groupées car les calculs doivent être faits individuellement pour chaque site.

Remarque : Même si les allocations de ressources pour le modèle actuel et les conditions ne sont pas nécessaires, il est important de les créer en gardant à l'esprit de futures conditions requises. Cette procédure permet d'éviter les frais généraux au cours développement futur du système.

Choisir le champ Horodatage

Comme indiqué précédemment, l'horodatage est très important dans la manière dont le Moteur de corrélation traite les Evénements. Les métriques calculent toujours le niveau de service par période, et les événements qui sont traités dans cette période sont ceux dont l'horodatage tombe dans la période.

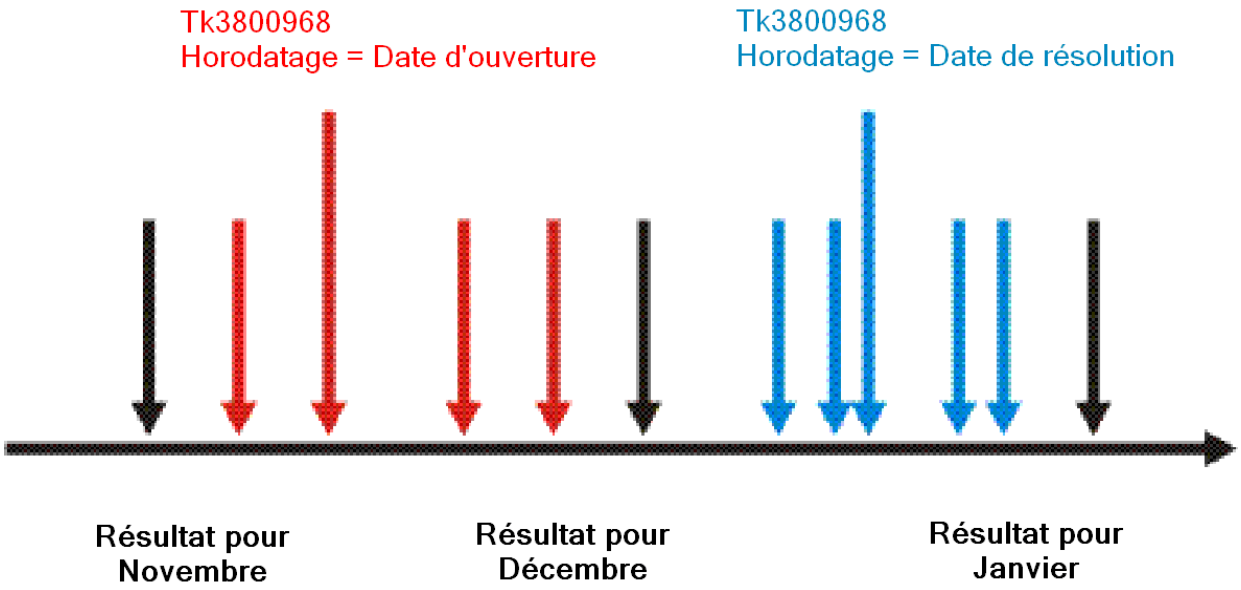
Dans la première étude de cas, la source de données n'avait qu'un champ de temps. Cette fois, le cas comporte trois champs pouvant être paramétrés comme horodatage. Examen des deux premiers enregistrements :

N° TK	Priorité TK.	Date d'ouverture	Date de fermeture	Date de résolution	Ref pers	Emplacement
3800968	1	19/12/2003 07:51	05/01/2004 11:31	22/12/2003 12:00	CM3	Londres
38000509	1	18/12/2003 09:21	05/01/2004 11:33	22/12/2003 12:00	CM1	Ipswitch

Le calcul du temps de résolution requiert le temps Date d'ouverture et Date de résolution. Pour les besoins de l'événement, il est possible de n''y attacher qu'un horodatage. Vous pouvez alors utiliser l'autre comme valeur dans les champs de valeur.

Si la valeur Date d'ouverture est utilisée comme horodatage, le ticket est inclus dans les résultats de décembre. Si la valeur Date de résolution est utilisée comme horodatage, alors le ticket est inclus dans les résultats de décembre. Les deux options sont viables. La sélection doit simplement indiquer où les tickets doivent s'afficher dans les rapports.

Il s'agit d'un point très important à considérer durant l'implémentation car il détermine quand les événements sont utilisés pour des calculs. Par exemple, si un ticket est levé en novembre mais n'est pas clôturé avant décembre, quand doit-il contribuer au résultat de SLA ? Est-il inséré dans les données de novembre ou de décembre ?



Dans ce cas, puisque que le ticket n'est signalé à la source de données qu'une fois clôturé, vous ne pouvez capturer les données qu'une fois le ticket clôturé. La date de clôture se situe généralement après les dates d'ouverture et de résolution. Dans le cas où la date d'ouverture a été choisie pour être l'horodatage, celui-ci doit être traité comme faisant partie des résultats de décembre. Le fait que la date de clôture était en janvier signifie que le ticket a été rapporté une fois décembre passé. Les résultats de décembre avaient donc déjà été publiés. Le moteur de corrélation remarque alors que l'événement est au passé car l'horodatage appartient à décembre, il déclenche ainsi de nouveaux calculs. Par conséquent, les résultats de décembre changent rétroactivement.

Ces conséquences doivent être parfaitement comprises afin de pouvoir définir quel champ de temps doit être choisi comme horodatage. Généralement, la date de clôture est choisie afin d'éviter que les rapports finaux ne changent rétroactivement.

En revanche, l'utilisation de la date de clôture comme Horodatage permet de disposer de davantage de temps pour entrer des calculs. Un ticket résolu ne peut être clôturé que bien plus tard.

Sachez toutefois que cette utilisation de la date de clôture pourrait également déclencher un processus dans l'organisation qui accélère le temps de clôture des tickets.

La dernière suggestion est alors :

Nom d'événement	Ticket de priorité 1 (peut également être défini pour d'autres priorités si nécessaire)
Comportement	Rapporté lorsque le ticket est clôturé
Champ Horodatage	Date de clôture
Champ Ressource	Champs Client + Site
Champ Type d'événement	Priorité
Champs Données	Tout
Attribut Type de ressource	Site du contractant
Allocation au contractant	Chaque site est alloué au contractant approprié
Allocation au service	Similaire à ci-dessus.
Allocation au groupe de ressources	Un groupe de ressources est créé pour chaque contractant afin de permettre le regroupement.

Enregistrement par	Pour les métriques groupées, l'enregistrement se fait par Ressource et la métrique est attachée à un groupe de ressources nommé Emplacements client CM3. L'enregistrement se fait par Contractant et Service pour les métriques de temps clôturées.

Exemple d'utilisation d'attributs personnalisés

L'étude de cas suivante présente un exemple pour plusieurs cibles dynamiques.

Etude de cas 9 : Cibles multiples dynamiques

Considérez ce scénario : tous les périphériques d'infrastructure matériel d'un environnement client disposent de cibles spécifiques définies pour leurs conditions de disponibilité. Utiliser l'approche de modélisation standard serait une tâche essentiellement difficile, celle-ci nécessiterait une gestion et énormément de groupement logique pour les périphériques dans l'utilisation du modèle de ressource. Outre la complexité, les cibles de ces périphériques peuvent changer avec le temps. Les détails étant stockés dans un CMDB externe, ces valeurs cibles sont mises à jour en CA Business Service Insight par un script de conversion. Pour l'exemple de script de conversion, consultez Exemples de meilleures pratiques pour les scripts de conversion (page 308).

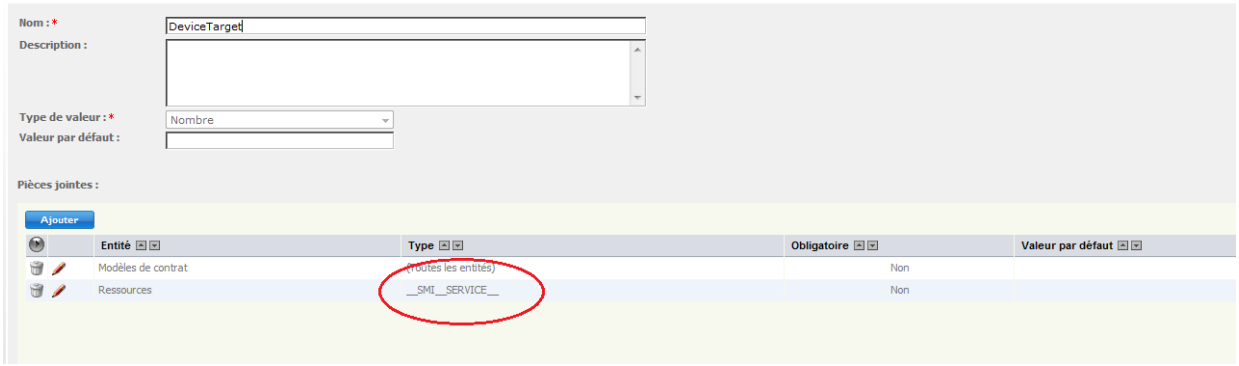
Dans ce cas, la métrique pourrait être comme suit :

% de disponibilité par périphérique matériel.

L'une des façons de modéliser efficacement ceci consiste à utiliser la fonctionnalité Attributs personnalisées en conjonction avec l'une des autres fonctionnalités clés, à savoir Cibles dynamiques. Les deux peuvent être utilisés avec une métrique groupée pour obtenir les résultats escomptés. L'ajout direct de la cible de niveau de service permet à la logique applicative de comparer le niveau de service de chaque ressource (périphérique matériel) à sa propre cible. Une métrique groupée fournit la conformité de service spécifique pour chaque pièce matériel n'utilisant qu'une seule métrique.

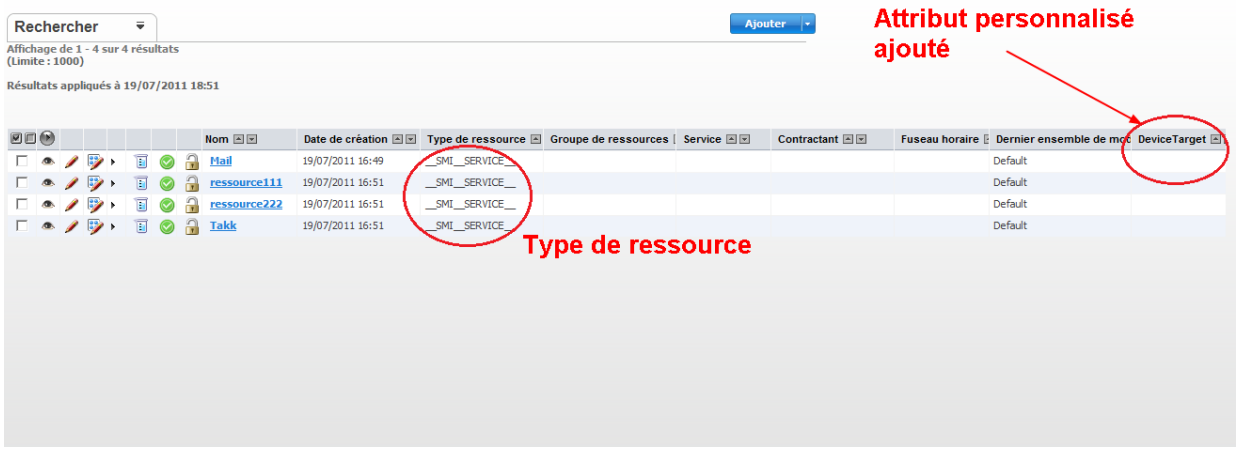
Par conséquent, il est nécessaire de créer d'abord l'attribut personnalisé en l'ajoutant au type de ressource de ces périphériques (où tous les périphériques sont une ressource de type Périphérique d'infrastructure). L'attribut personnalisé créé est appelé DeviceTarget et peut être ajouté depuis le menu dans *Catalogue des services* > *Attributs personnalisés*. Remarque : Lors de la création de l'attribut personnalisé, vous devez le lier au(x) type(s) de ressource qui l'exigent.

Détails de l'attribut personnalisé



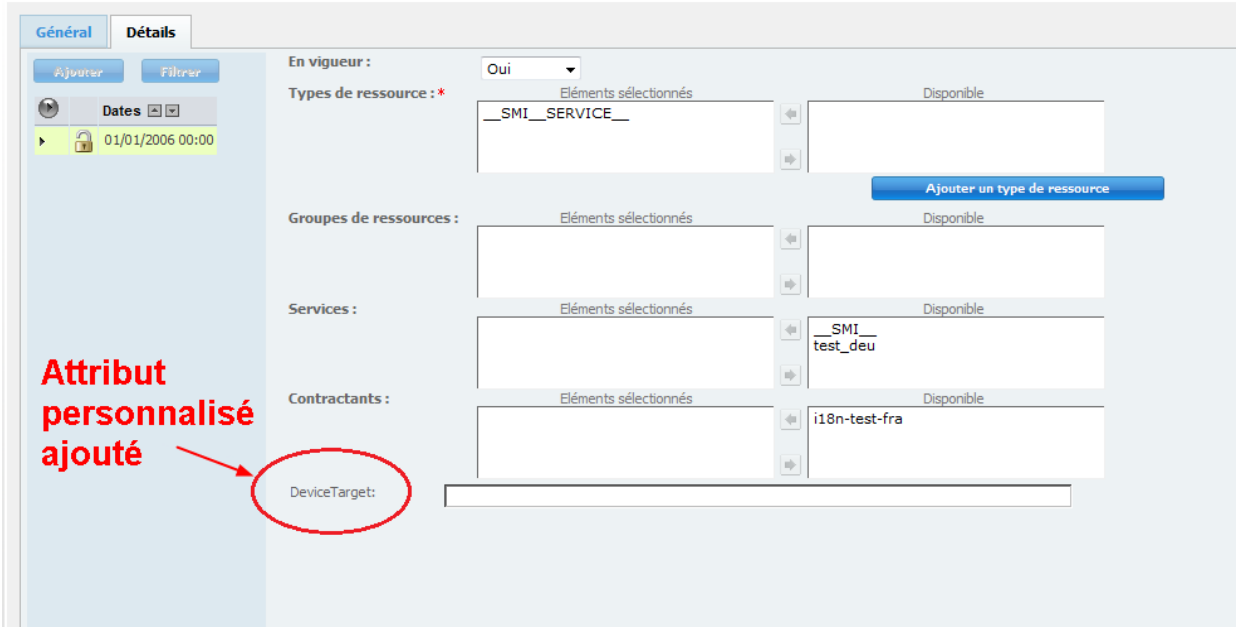
Il est désormais possible de voir dans les ressources du système le nouvel attribut personnalisé disponible pour le type de ressource auquel il était lié.

Ressources



Et les ressources distinctes disposent d'un nouveau champ que vous pouvez mettre à jour.

Détails de la ressource



Dans cet exemple, ce champ serait normalement inséré/mis à jour par le script de conversion.

Maintenant que chacune des ressources dispose d'une cible spécifiée, vous pouvez développer la logique pour effectuer le calcul requis (après avoir soumis les changements de ressources). L'échantillon de code suivant vous montre comment extraire la valeur d'attribut personnalisée de la ressource (en gras).

```
Option Explicit

Dim TotalTime
Dim OutageTime
Dim PeriodStart

Sub OnRegistration(dispatcher)
   dispatcher.RegisterByResource "OnDeviceOutageEvent", "DeviceOutageEvent", _
      Context.ClusterItem
End sub

Sub OnLoad(TIME)
   TotalTime = 0
   OutageTime = 0
End sub

Sub OnPeriodStart(TIME)
   TotalTime = 0
   OutageTime = 0
   PeriodStart = TIME
End sub

Sub OnPeriodEnd(TIME, isComplete)
   TotalTime = Tools.NetTime(PeriodStart, TIME)
End sub

Sub OnDeviceOutageEvent(eventDetails)
   OutageTime = OutageTime + Tools.NetTime (eventDetails ("OutageStartTime"), _
      eventDetails ("OutageEndTime"))
End sub

Function Target
   Target = eventDetails.CustomAttribute ("DeviceTarget")
End Function

Function Result
   If TotalTime > 0 Then
      Result = (TotalTime - OutageTime)/TotalTime
   Else
      Result = Null
   End If
End Function
```

Exemples de script de conversion

Les antécédents suivants incluent des exemples de mises à jour de modèle de conversion et de ressource automatiques de base.

Etude de cas 10 : Conversion automatique de base

L'exemple de script de conversion fourni ici est un simple processus de traitement des entrées En attente dans l'écran Entrées de conversion.

L'agent OnTranslationEvent réalise un simple contrôle sur le premier caractère de la ressource et accomplit une action selon la valeur : si cette dernière est "a", l'entrée de conversion est paramétrée sur "ignorer", si elle est "b", l'entrée est supprimée, si elle est "c", elle est convertie, dans le cas contraire elle restera inchangée pour être convertie manuellement. Notez que par le biais du code, les compteurs gardent la trace des actions accomplies pendant l'exécution de script. Ceci est très utile pour déboguer et documenter les exécutions de script chaque fois qu'il est exécuté, particulièrement lorsque le script est automatisé. Il est très important de se souvenir de la commande *Tools.Commit* des dernières fonctions ; sans elle, aucun des changements apportés par le script ne sera enregistré dans la base de données.

La fonction TranslateResource() appelée vérifie simplement si la ressource du même nom que celle passée par l'entrée de conversion en attente (avec le préfixe E2E-) existe dans le système. Si ce n'est pas le cas, le script ajoute cette ressource, puis procède à la conversion. Si elle existe déjà, alors une entrée de conversion est créée depuis la chaîne de ressource vers la ressource CA Business Service Insight existante.

La dernière fonction de Résultat du script consiste simplement à sortir une description des tâches accomplies par le script. Le code est le suivant :

```
Option Explicit

dim translated
dim ignored
dim deleted
dim manually
dim ActionDate

Sub OnLoad()
   'tools.log "Translation Script: In OnLoad procedure", "I"
End sub

Sub OnTranslationEvent(entryDetails)
   Dim dump
   dump = entryDetails.Dump
   tools.log dump

   Dim resource, entryId
   entryId = entryDetails.EntryId
   resource = entryDetails.FieldValue(1)
   ActionDate = entryDetails.LastActionDate

   If mid(resource,1,1) = "a" Then
      tools.IgnoreEntry entryId
      ignored = ignored + 1
      tools.log "ignored" & entryId & " " & resource
   Else If mid(resource,1,1) = "b" Then
      tools.DeleteEntry entryId
      deleted = deleted + 1
      tools.log "deleted" & entryId & " " & resource
   Else If mid(resource,1,1) = "c" Then
      TranslateResource resource, entryId
      tools.log "translated" & entryId & " " & resource
   Else
      tools.SetManualTranslationEntry entryId, 1
      manually = manually + 1
      tools.log "manually" & entryId & " " & resource
   End if

   Tools.commit
End sub

Sub TranslateResource(resource, entryId)
   Dim newName
   Dim vector
   newName = "E2E-" & resource
```

```
   if NOT tools.IsResourceExists(newName) Then
      Dim resourceDetails
      set resourceDetails = tools.GetResourceDetailsByDate(resource,ActionDate)
      resourceDetails("ResourceName") = newName
      resourceDetails("ResourceTypes") = "E2E Transactions"
      tools.AddResourceByMap resourceDetails
   end if

   tools.TranslateEntry entryId, newName
End sub


Sub Main()
End Sub


Function Result
   Result = translated & "entries were translated, "& _
      ignored & "entries were ignored," & _
      deleted & "entries were deleted and "& _
      manually & "entries were set to manually update!"
End Function
```

Etude de cas 11 : Mises à jour du modèle de ressource

Une autre utilisation des scripts de conversion consiste à mettre à jour le modèle de ressource CA Business Service Insight avec des valeurs prises d'une source de données externe. Bien qu'il ne soit plus lié à la conversion, cet exemple est une aide précieuse pour mettre à jour le système de façon automatique.

La section précédente sur les Attributs personnalisés décrivait un scénario où les périphériques d'infrastructure matérielle d'une organisation sont stockés dans un CMDB externe, avec une cible de disponibilité attendue pour chaque périphérique. Ces informations ont besoin d'être répliquées dans le modèle de ressource CA Business Service Insight pour garder à jour le mappage d'infrastructure (et les cibles de périphérique).

Dans ce cas, le script est requis pour accomplir les tâches suivantes :

- Ajouter tout nouveau périphérique matériel qui n'existe actuellement pas dans le système.
- Mettre à jour la cible de disponibilité attendue des périphériques existants (si la cible est changée).

Le script est le suivant :

```
Option Explicit

'*********************************************************************
'Variables globales et constantes
'*********************************************************************

Dim added
Dim updated
Dim ChangeSetName

added = 0
updated = 0

Const RESOURCE_TYPE = "Infrastructure Devices"
Const RESOURCE_GROUP = "InfraServers"
Const CHANGESET_NAME = "Infrastructure Devices"
Const CHANGESET_EFFECTIVE_DATE = "01/01/2007"

'*********************************************************************
'Sub OnLoad :
'Préparation des entités d'infrastructure fondatrices
'*********************************************************************

Sub OnLoad()
```

```
   Tools.log "Translation Script : In OnLoad procedure", "D"

   'Cherche une version de ressource préliminaire existante
   Dim ChangeSetMap
   Set ChangeSetMap=Tools.SearchChangeSets(CHANGESET_EFFECTIVE_DATE,
CHANGESET_NAME)

   'Si aucune version existante, créer une nouvelle version
   If ChangeSetMap.EMPTY Then
      Tools.AddChangeSet CHANGESET_NAME, CHANGESET_EFFECTIVE_DATE, CHANGESET_NAME
      Tools.Log "Changes set '" & CHANGESET_NAME & "' added."
   End If

   Set ChangeSetMap = Nothing
End sub

Sub OnTranslationEvent(EntryDetails)
End sub

Sub Main()
   Dim conn, rs, resource, deviceTarget, resource_id, resMap, custAttrib,
custAttribValue
   Set conn = CreateObject("ADODB.Connection")
   Set rs = CreateObject("ADODB.RecordSet")
   conn.open "DSN=HardwareDevices"
   rs.Open "select * from tblServers", conn

   Do While Not rs.EOF
      resource = rs("serverName")
      deviceTarget= rs("DeviceTarget")
      'Ajoute des ressources à la dernière version si cela n'est pas déjà fait
      If NOT Tools.IsResourceExists(resource) Then
         resource_id = Tools.AddResource(resource, CHANGESET_NAME,
"Infrastructure Device", RESOURCE_TYPE, RESOURCE_GROUP, Null, Null)
         Tools.UpdateResourcesCustomAttribute resource, CHANGESET_NAME,
"DeviceTarget", deviceTarget
         Tools.Log "AddingResource: " & resource & " with target: " &
deviceTarget & " ; assigned ID= " & resource_id
         added = added + 1
      Else
         Set resMap = Tools.GetResourceDetails(resource,CHANGESET_NAME, False)
         Set custAttrib = resMap("CustomAttributes")
         custAttribValue =
CDbl(custAttrib("DeviceTarget")"(CustomAttributeValue"))
         If CDbl(deviceTarget) <> custAttribValue Then
            Tools.UpdateResourcesCustomAttribute resource, CHANGESET_NAME,
"DeviceTarget", deviceTarget
            Tools.Log "Updating Resource target for : " & resource & " from: " &
deviceTarget & " to " & custAttribValue
```

```
            updated = updated + 1
         End If
      End If

      Tools.Commit
      rs.MoveNext
   Loop

   rs.Close
   conn.Close
   Set rs = Nothing
   Set conn = Nothing
End sub

Function Result
   ' Valide la transaction
   Tools.CommitChangeSets CHANGESET_NAME
   Result = added & " resources added and " & updated & " resources updated."
End Function

'*******************************************************************************
```

Exemples d'écriture de scripts de logique applicative

Voici quelques directives générales sur l'écriture de scripts de logique applicative :

Variables globales

- Veillez à compléter les valeurs globales que vous déclarez. Le mécanisme d'état PSL ne peut pas enregistrer de variables comportant des valeurs nulles

Codage général

- Avant d'utiliser un des objets répertoriés ci-dessous, vérifiez qu'il existe (à l'aide d'une méthode IsExist appropriée) :
  - Tous les types de paramètre (par ex. table, liste, etc.)
  - Attribut personnalisé
  - Ressource
- Veillez à fournir un module de logique applicative comportant tous les paramètres requis
- Avant de changer le nom d'une ressource, vérifiez quelles métriques l'utilisent et mettez-les à jour en conséquence

Plans

- Utilisation des plans dans les métriques groupées : Les plans exigent davantage de calcul du moteur, sachez qu'un groupe sur une métrique multipliera l'effort par le nombre d'éléments groupés.
- Les grands plans globaux de la logique applicative des métriques groupées ne doivent être qu'utilisées qu'après mûre considération. Lorsque que le moteur calcule une métrique groupée, il est occupé à charger séparément les variables globales des états pour chaque élément.
- Veillez à effacer les plans et vecteurs après avoir terminé.
- Si l'utilisation de grands plans est nécessaire, assurez-vous de les gérer efficacement en les divisant en plages logiques.

Etude de cas 12 : Utilisation de la logique de compteur pour calculer le nombre de défaillances

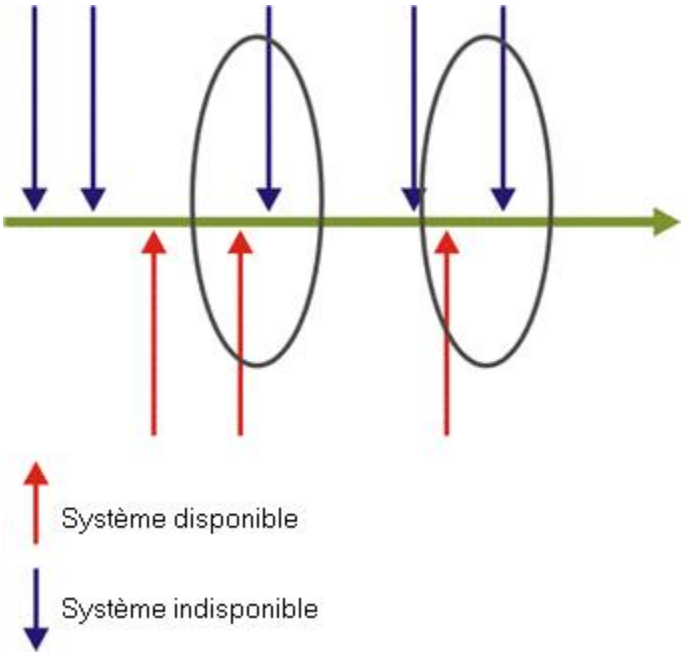
L'exemple suivant calcule le nombre de défaillances qui se sont produits dans une période de calcul donnée. Vous pouvez considérer cette formule et les méthodes utilisées pour l'implémenter, comme l'exemple d'une formule exigée lorsqu'il est nécessaire d'effectuer un calcul.

Les suppositions de calcul suivantes sont utilisées :

- Evénements d'entrée :
  - Evénement de disponibilité, Statut (0/1)
  - L'événement de disponibilité se produit selon des intervalles de quelques minutes. Il est impossible de prévoir la fréquence des événements (l'événement peut se produire toutes les cinq minutes, ou seulement une fois une heure), il peut également y avoir des événements redondants (exemple : 0, 0, 0, 1, 1, 0, etc).
- Période d'application : heure d'ouverture, les défaillances survenant hors de la période d'application ne sont pas prises en compte.
- Le résultat obligatoire est le nombre de défaillances survenues durant cette période.

Afin de dénombrer les défaillances survenues durant la période de calcul, il est nécessaire de stocker une variable de compteur périodique ainsi qu'une variable stockant le statut du système. Etant donné que des informations d'événement redondantes peuvent être reçues (c.-à-d. un événement Up suivi d'un autre événement Up), il est nécessaire de dénombrer le nombre d'emplacements où un changement de statut du système de Up à Down est survenu, sans compter chaque événement Down reçu, ceux-ci pouvant être redondants et représenter des défaillances déjà prises en compte.

L'illustration suivante décrit graphiquement le compteur des temps Up et Down du système.



Points importants devant être considérés :

- **Constantes** : l'utilisation de définitions constantes dans le code plutôt que de valeurs constantes est recommandée. De cette façon, il est plus aisé de changer la valeur dans la définition constante que de devoir la chercher dans le code entier. Il est également facile de la changer en une utilisation de paramètre, si nécessaire. Dans l'exemple susmentionné, les valeurs qui représentent le statut du système reçu dans l'événement sont définies comme des constantes afin de rendre le code plus compréhensible, mais également pour distinguer lorsque le numéro zéro est utilisé comme nombre de calcul plutôt que représentant du statut du système.
- **Variables globales** :
  - **Compteur** : la définition de la variable de compteur est une définition globale. Dans la formule, elle est déclarée au sommet du code et à l'extérieur de toute fonction ou sous-routine. Il est important dans ce cas de définir la variable de compteur comme variable globale. Cela permet d'une part de l'utiliser dans plusieurs fonctions/sous-routines à travers le code et d'autre part au système de garder sa valeur durant la période de calcul et de fournir son résultat en fin de période.

Dans cet exemple, la variable de compteur doit être utilisée à trois endroits dans le code :

- Initialisation au début d'une période.
- Avancement dans le cas d'un événement échoué dans le gestionnaire d'événements.
- Sortie par la fonction de résultat à la fin de la période.

– **Statut précédent du système** : cette variable contient le statut du système et est mise à jour lorsqu'un événement comportant le statut du système est reçu. Cette variable doit également être globale, car elle est utilisée dans plusieurs fonctions/sous-routines du code, par exemple :

- Initialisation au début du calcul
- Mise à jour lors de la réception d'un nouvel événement

- **Remarques concernant la période d'application :** il est possible d'utiliser la valeur de la propriété *context.IsWithinTimeSlot* afin de vérifier si une défaillance survient dans une période d'application. Le contexte est un objet global que vous pouvez utiliser n'importe où dans le code, et dans ce cas est utilisée pour indiquer lorsqu'une défaillance est reçue et si elle se trouve à l'intérieur ou à l'extérieur de la période d'application. Si à l'horodatage de l'événement l'indicateur retourne VRAI, alors l'événement se produit dans une période d'application. S'il retourne FAUX, cela indique que l'événement a eu lieu à l'extérieur de la période d'application. D'après la logique requise, une défaillance qui se produit à l'extérieur de la période d'application est ignorée et par conséquent non prise en compte.
- **Initialisez les variables** :

– **Variable du compteur** : contient une valeur périodique et est de fait initialisée dans l'agent *OnPeriodstart* pour s'assurer que chaque période de calcul dénombre ses défaillances séparément. Chaque période commence par zéro et les sorties uniquement par le nombre de défaillances de cette période.

Dans les cas où calculer les défaillances accumulées est requis pour chaque période (signifiant que le résultat de chaque période est égal à toutes les défaillances qui se sont produites jusqu'à la fin de cette période, y compris toutes les périodes avant cela), il est indispensable d'initialiser le compteur uniquement dans l'agent *OnLoad* et de le supprimer de l'agent *OnPeriodStart*. Ainsi, le compteur continue de dénombrer et d'accumuler entre les périodes, comme affiché ci-dessous :

```
Sub OnLoad(time)
    FingerprInted=0
End sub
```

- **Variable de statut du système** : ne doit être initialisé qu'une fois au début du calcul, puis être mis à jour à chaque événement de statut. Cette variable conserve sa valeur tout au long du calcul et n'est pas restreinte à une période de calcul. Elle doit également conserver sa valeur entre les périodes de calcul. Car le statut du serveur est inconnu avant de recevoir le premier événement, une supposition doit être faite quant au statut de système. La supposition usuelle est que le système soit "Up". Cette supposition doit être convenue et vérifiée car celle-ci peut mener à la situation suivante :

  Si le statut du système se trouvait être Down lors du calcul et que le premier événement arrivant rapporte ce statut, une défaillance résulterait du fait que le statut Up était supposé. Cette défaillance serait comptée dans la première période de calcul et ce même si elle n'est pas nécessairement survenu durant cette celle-ci.

```
Option Explicit

'Définitions de constantes
Const UP=1
Const DOWN=0

'Variable globale pour les défaillances de calcul
Dim FingerprInted
Dim SystemStatus

Sub OnRegistration(dispatcher)
   ' Les paramètres de la méthode sont : <procedure name>, <Event Type>
   dispatcher.RegisterByContractPartyAndService
"OnAvailabilityEvent","AvailabilityEvent"
End sub

Sub OnLoad(time)
   SystemStatus = UP 'assume le premier statut du système
End sub

Sub OnPeriodStart(time)
   FingerprInted = 0
End sub

Sub OnAvailabilityEvent(eventDetails)
   ' Compté dans le cas où c'est une défaillance et dans la période d'application
   If context.IsWithinTimeSlot and SystemStatus=UP and _
         eventDetails("Status")=DOWN Then
      FingerprInted = FingerprInted + 1
   End If
   ' met à jour le statut du système pour la comparaison suivante
   SystemStatus = eventDetails("Status")
End sub
```

```
Function Result
   Result = FingerprInted
End Function
```

Etude de cas 13 : Traitement du groupe de composant dynamique

Il est souvent obligatoire de stocker des valeurs pour un groupe de ressources dans lequel les membres peuvent être dynamiques et changer durant la période de calcul. Dans l'exemple de condition de calcul suivant, il est nécessaire de réaliser un calcul intermédiaire pour chacune des ressources afin d'atteindre le résultat total final.

Quelques autres exemples de ce type :

- **Incidents sur site** : calcule le pourcentage d'emplacements comportant des incidents résolus en plus de X heures, ou dénombre les emplacements ayant une moyenne de résolution des incidents supérieure à X heures.

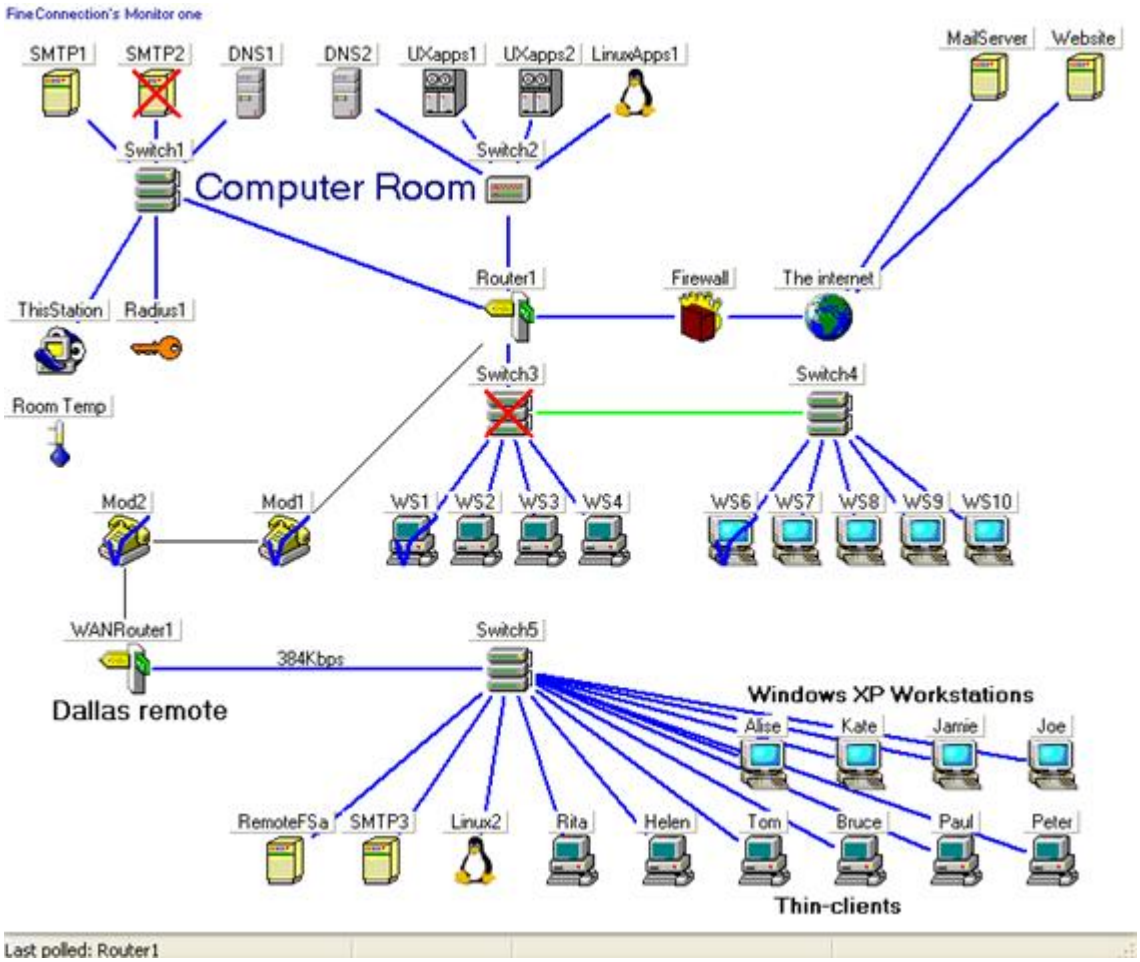
  Dans ces exemples, les sites sont des ressources ayant des incidents associés.

- **Disponibilité du serveur :** recense le nombre de serveurs dont le temps de disponibilité était plus grand que X%.

  Un serveur est une ressource pour laquelle le pourcentage de disponibilité doit être évalué.

- **Types de transactions** : calcule le pourcentage de type de transactions qui échoue plus de X fois.

  Dans ce cas, un type de transaction est une ressource ayant des événements de défaillance associée. Pour chaque type de transaction, un compteur de défaillance est stocké comme un résultat intermédiaire tandis que le nombre de types de transaction différents possédant plus de X défaillances sont dénombrés.


Fine Connection's Monitor one
SMTP1
SMTP2
DNS1
DNS2
UXapps1
UXapps2
LinuxApps1
MailServer
Website
Switch1
Switch2
Computer Room
Router1
Firewall
The internet
ThisStation
Radius1
Switch3
Switch4
Room Temp
Mod2
Mod1
WS1
WS2
WS3
WS4
WS6
WS7
WS8
WS9
WS10
WANRouter1
384Kbps
Switch5
Dallas remote
Windows XP Workstations
Alise
Kate
Jamie
Joe
RemoteFSa
SMTP3
Linux2
Rita
Helen
Tom
Bruce
Paul
Peter
Thin-clients
Last polled: Router1

***Exemple* :**

Pour la condition de calcul suivante :

Calculer le pourcentage de disponibilité pour un système composé d'un groupement de serveurs. Le système est considéré disponible uniquement lorsque tous les serveurs sous-jacents sont disponibles.

La solution sera implémentée comme suit :

La disponibilité de système est évaluée par la disponibilité de ses ressources groupées sous-jacentes. Dans ce cas, il est nécessaire de gérer et stocker le statut de toutes les ressources groupées pour évaluer le statut du système à chaque point dans le temps.

Pour ce faire, la formule doit contenir un plan (le plan ServerAvailabilityIndicators dans l'exemple de code ci-dessus) ayant une entrée pour chacune des ressources surveillées. Vu que tous les objets du plan requièrent un champ clé pour référencer la valeur associée, ce plan aura l'indicateur de ressource comme clé (ID de ressource) et la valeur sera le statut du composant. Quand le statut d'un composant change, l'entrée appropriée dans ce plan doit également être modifiée. La formule mettra à jour le statut de la ressource appropriée dans le plan pour chaque événement de disponibilité, et réévaluera le statut du système en conséquence.

Puisque les ressources surveillées peuvent changer (certaines peuvent être ajoutées ou supprimées durant la période de calcul), l'ajout d'une fonction dans la formule doit le gérer. Celle-ci aura pour but d'identifier le changement et de mettre à jour le plan de composant surveillé par l'ajout d'une nouvelle entrée par nouveau composant, ou par la suppression d'une entrée si le composant a été supprimé.

OnRegistration est le gestionnaire d'événements qui gère un événement d'enregistrement déclenché lors d'un changement dans les ressources surveillées. Un tel changement peut se produire lors de la validation d'une ressource (ou d'un ensemble de modifications) pouvant produire des changements dans les ressources incluses dans le calcul, d'après la méthode d'enregistrement de la formule.

Il est nécessaire de mettre à jour le plan contenant les statuts de ressource durant chaque enregistrement et pour tout changement requis. Cela signifie comparer le plan utilisé auparavant pour contenir les statuts de ressources et celui utilisé pour contenir les ressources au moment de l'enregistrement (d'après la méthode d'enregistrement), puis inclure toutes les ressources ajoutées ou effacer les ressources supprimées.

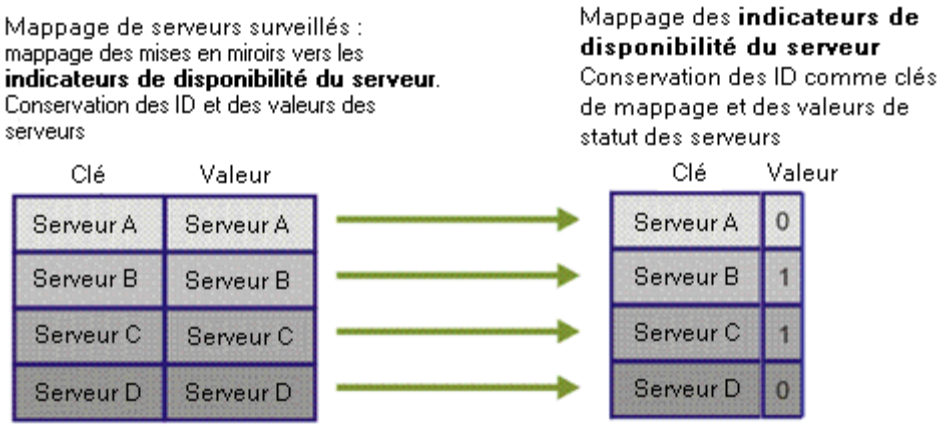
La procédure OnRegistration doit alors exécuter une fonction comparant les ressources surveillées aux nouvelles ressources allouées, ceci afin de structurer les ressources surveillées en conséquence.

L'objet Contexte dispose d'un ensemble de méthodes mettant en parallèle les méthodes d'enregistrement. Cela renvoie un plan comportant toute ressource faisant partie des ressources d'après la méthode d'enregistrement.

Dans l'exemple suivant, l'enregistrement de la formule est ByContractParty, et la même méthode est par conséquent utilisée par Context.ResourcesOfContractParty. Cela renvoie un plan comportant toute ressource faisant partie de cet ensemble au moment de l'enregistrement.

Il est nécessaire d'itérer en parallèle à travers les plans pour comparer les deux plans. Le plans sont itérés en utilisant l'instruction "For Each". Cette instruction permet l'itération à travers les valeurs d'un plan, par conséquent, un second plan miroir du premier est nécessaire pour pouvoir itérer à travers les ressources et non leurs valeurs de statut. Cette instruction répète les valeurs d'un plan et non ses clés. Par conséquent, un plan supplémentaire comportant les ID en tant que clé et valeur est nécessaire. Maintenez également le plan miroir pour refléter continuellement le plan de statuts afin que les deux contiennent en permanence le même ensemble de ressources. Cela signifie que lorsque le plan de statuts est mis à jour, le plan miroir doit l'être également.

L'illustration suivante affiche le concept des plans miroirs.



Exemple **:**

```
Dim ServerAvailabilityIndicators
Dim MonitoredServers
Set ServerAvailabilityIndicators=CreateObject("SlalomMap.Map")
Set MonitoredServers=CreateObject("SlalomMap.Map")

Sub OnRegistration(dispatcher)
   dispatcher.RegisterByContractParty "OnAvailabilityEvent",_
      "Availability Event", "SAP Production Server"
   Dim AllocatedServers
   Set AllocatedServers = Context.ResourcesOfContractParty("SAP Production
Server")
   UpdateMonitoredServers AllocatedServers
End sub

Sub UpdateMonitoredServers(allocatedServers)
   Dim Server
   For Each Server In allocatedServers
      If Not MonitoredServers.Exist(Server) Then
         MonitoredServers(Server) = Server
         ServerAvailabilityIndicators(Server)=True
      End If
   Next

   For Each Server In MonitoredServers
      If Not allocatedServers.Exist(Server) Then
         MonitoredServers.Erase Server
         ServerAvailabilityIndicators.Erase Server
      End If
   Next
End sub
```

Exemple **:**

La fonction suivante démontre la manière dont le plan miroir est utilisé à travers le plan de statuts lorsque l'évaluation du statut du système entier est requise, évaluation basée sur le statut de chaque ressource surveillée.

Dans cet exemple, le système est considéré disponible si toutes les ressources sont disponibles. Avoir un seul composant éteint est assez pour considérer le système éteint :

```
Function SystemAvailability
   Dim Server
   For Each Server In MonitoredServers
      If ServerAvailabilityIndicators(Server) = DOWN then
         SystemAvailability=DOWN
         Exit Function
      End if
   Next
```

```
End Function
```

Un exemple de logique applicative complet avec traitement de ressource dynamique est décrit dans l'exemple de modèle de conception suivant.

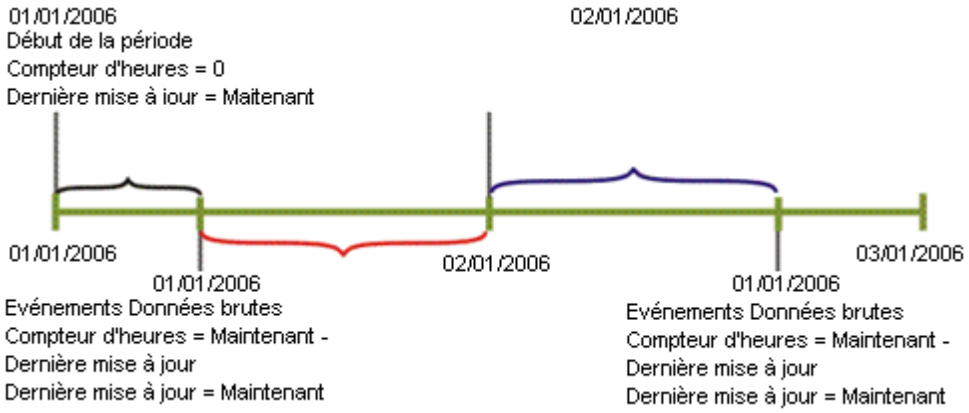
Cas d'étude 14 : Traitement de l'horloge d'accumulation de temps

Le modèle de conception décrit dans cette section est adapté lorsque le résultat requis est une fonction d'une période de temps écoulé entre des événements, par exemple :

- Temps disponible calculé sur le temps écoulé entre deux défaillances.
- Temps de résolution calculé sur le temps entre l'ouverture et la fermeture de l'incident.

Afin de cumuler du temps, il est nécessaire d'assigner une variable dans laquelle le temps peut être cumulé (en secondes), et d'implémenter une fonction qui vérifie les conditions et le temps accumulés depuis la dernière mise à jour. Cette fonction est alors exécutée pour tous les événements reçus dans la formule.

L'illustration suivante décrit l'horloge traitant l'accumulation de temps.



La variable LastUpdateTime stocke la date de la dernière mise à jour, peu importe si le compteur de temps était mis à jour. La fonction contient la condition déterminant si le temps doit être mis à jour et cumulé. Par exemple, le temps ne doit pas être considéré si : il dépasse la période d'application, le statut du système était Down, l'incident avait un statut en attente.

Bien que la situation détaillée ici utilise souvent la fonction Tools.NetTime() pour calculer des durées, la fonction VB standard DateDiff() peut être préférable dans certains cas.

La fonction Tools.NetTime engendre des frais généraux dans la vérification de la période d'application chaque fois qu'elle est utilisée. Il est recommandé d'éviter NetTime pour les procédures d'événement de données car celles-ci sont appelées pour tout nouvel événement arrivant, invoquant ainsi l'appel NetTime. Si votre période d'application est 24/7, il vous est recommandé d'utiliser la fonction DateDiff pour éviter les frais généraux lors de la vérification de la période d'application.

Exemple 1 **:**

La routine de mise à jours des compteurs suivante cumule la période totale de temps dans la variable de PeriodNetTime. La routine AvailabilityTime cumule le temps pendant lequel le statut du système était Up, signifiant que le système était disponible.

L'objet Outils contient la méthode NetTime, NetTime(beginTime, endTime)0. Cette méthode renvoie le nombre de secondes entre beginTime et endTime se trouvant dans la période d'application de la métrique actuelle. Tout temps entre ces deux variables fait partie d'une période d'application exclue, cette procédure est par conséquent très utilisée pour les calculs de durée où une période d'application est utilisée. (Par exemple : pour des tickets de priorité 1 résolus dans les quatre heures ouvertes, bien qu'un ticket envoyé en fin de journée ne sera pas résolu avant le lendemain matin, celui-ci reste dans le SLA à cause de l'exclusion des heures de période d'application.)

```
Sub UpdateCounters (time)
   PeriodNetTime = PeriodNetTime + tools.NetTime (LastUpdateTime, time)
   If SystemStatus = UP Then
      AvailabilityTime = AvailabilityTime + tools.NetTime (LastUpdateTime, time)
   End If
   LastUpdateTime = time
End sub
```

Exemple 2 **:**

L'exemple suivant calcule la disponibilité d'application en gérant les événements Up et Down de plusieurs composants critiques, ainsi que les événements de maintenance de ces composants. Si tous les composants sont sous maintenance, le temps n'est pas considéré comme un temps de disponibilité attendu.

La sous-routine UpdateCounters avance le compteur de temps si nécessaire et est exécutée avec tous les événements reçus dans la formule (événement de données brutes/événement de moteur). Elle met également à jour le temps de disponibilité attendu dans les cas où le temps est dans la période d'application et où les composants ne sont pas dans une période d'indisponibilité planifiée. La formule ne met à jour le temps de disponibilité actuel que lorsqu'elle dispose d'un statut de système Up.

DateDiff est une fonction standard de VB qui renvoie le temps entre deux dates mais n'exclut aucune information de période d'application.

```
'Force la déclaration de variable
Option Explicit

'Variables globales
Dim ExpectedAvailabilityTime
Dim ActualAvailabilityTime
Dim LastUpdateTime
Dim AvailabilityIndicators
Dim MonitoredComponents
Dim DowntimeStatuses

'Création d'objets Plans
Set AvailabilityIndicators=CreateObject("SlalomMap.Map")
Set MonitoredComponents=CreateObject("SlalomMap.Map")
Set DowntimeStatuses=CreateObject("SlalomMap.Map")

'Après le chargement et chaque fois que survient un changement d'infrastructure
Sub OnRegistration(dispatcher)
   dispatcher.RegisterByResourceGroup "OnComponentDownEvent","Component
Down","Application Components"
   dispatcher.RegisterByResourceGroup "OnComponentUpEvent","Component
Up","Application Components"
   dispatcher.RegisterByResourceGroup "OnMaintenanceStartEvent","Maintenance
Start","Application Components"
   dispatcher.RegisterByResourceGroup "OnMaintenanceEndEvent","Maintenance
End","Application Components"
   UpdateCounters Context.RegistrationTime

   Dim AllocatedComponents
   Set AllocatedComponents = Context.ResourcesOfResourceGroup("Application
Components")
```

```
   's'assure que la formule ne surveille que toutes les ressources pertinentes
   UpdateMonitoredComponents AllocatedComponents
End sub

Sub OnLoad(time)
   'Lorsque le système démarre pour la première fois - part du principe que la
disponibilité est OK
   LastUpdateTime = time
End sub

Sub OnPeriodStart(time)
   'initialise les compteurs pour renouveler le calcul périodique
   ExpectedAvailabilityTime = 0
   ActualAvailabilityTime = 0
End sub

Sub OnTimeslotEnter(time)
   UpdateCounters time
End sub

Sub OnTimeslotExit(time)
   UpdateCounters time
End sub

Sub OnComponentDownEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'écrire le statut de disponibilité de la ressource faisant l'objet d'un
rapport
   AvailabilityIndicators(eventDetails.ResourceId) = _
      AvailabilityIndicators(eventDetails.ResourceId)+1
End sub

Sub OnComponentUpEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'écrire le statut de disponibilité de la ressource faisant l'objet d'un
rapport
   AvailabilityIndicators(eventDetails.ResourceId)= _
      AvailabilityIndicators(eventDetails.ResourceId)-1
End sub

Sub OnMaintenanceStartEvent(eventDetails)
   UpdateCounters eventDetails.Time
   'écrire le statut de disponibilité de la ressource faisant l'objet d'un
rapport
   DowntimeStatuses(eventDetails.ResourceId) = _
      DowntimeStatuses(eventDetails.ResourceId)+1
End sub

Sub OnMaintenanceEndEvent(eventDetails)
```

```
   UpdateCounters eventDetails.Time
   'écrire le statut de disponibilité de la ressource faisant l'objet d'un
rapport
   DowntimeStatuses(eventDetails.ResourceId) = _
      DowntimeStatuses(eventDetails.ResourceId)-1
End sub

Sub OnPeriodEnd(time,isComplete)
   UpdateCounters time
End sub

Function Result
   If ExpectedAvailabilityTime <> 0 Then
      Résultat = 100 * (ActualAvailabilityTime/ExpectedAvailabilityTime)
   Else
      Result = Null
   End If
End Function

Sub UpdateCounters(time)
   If Context.IsWithinTimeslot And Not AllComponentsAreInPlannedDowntime Then
      'met à jour le compteur de secondes dans la période (lorsque la
disponibilité est attendue)
      ExpectedAvailabilityTime = ExpectedAvailabilityTime +
DateDiff("s",LastUpdateTime,time)
      If SystemIsAvailable Then
         'met à jour le compteur des seconde de disponibilité
         ActualAvailabilityTime = ActualAvailabilityTime +
DateDiff("s",LastUpdateTime,time)
      End If
   End If
   LastUpdateTime=time
End sub

Sub UpdateMonitoredComponents(allocatedComponents)
   Dim Component
   'ajoute au plan des composants surveillés tous les nouveaux composants à
surveiller
   For Each Component In allocatedComponents
      If Not MonitoredComponents.Exist(Component) Then
         MonitoredComponents(Component) = Component
         AvailabilityIndicators(Component) = 0
         DowntimeStatuses(Component) = 0
      End If
   Next

   'supprime du plan des composants surveillés tous les composants devenus non
pertinents
   For Each Component In MonitoredComponents
```

```
      If Not allocatedComponents.Exist(Component) Then
         MonitoredComponents.Erase Component
         AvailabilityIndicators.Erase Component
         DowntimeStatuses.Erase Component
      End If
   Next
End sub

Function SystemIsAvailable
   Dim SystemAvailability
   SystemAvailability = True

   Dim Component
   Dim ComponentAvailability
   For Each Component In MonitoredComponents
      ComponentAvailability = AvailabilityIndicators(Component) = 0 _
         Or DowntimeStatuses(Component) > 0
      'la disponibilité du système est évaluée avec la disponibilité
      SystemAvailability = SystemAvailability And ComponentAvailability
   Next
   SystemIsAvailable = SystemAvailability
End Function

Function AllComponentsAreInPlannedDowntime
   Dim ComponentsInPlannedDowntime
   ComponentsInPlannedDowntime = 0
   Dim Component
   For Each Component In MonitoredComponents
      If DowntimeStatuses(Component) > 0 Then
         ComponentsInPlannedDowntime = ComponentsInPlannedDowntime + 1
      End If
   Next

   If ComponentsInPlannedDowntime = MonitoredComponents.Count Then
      AllComponentsAreInPlannedDowntime = True
   Else
      AllComponentsAreInPlannedDowntime = False
   End If
End Function
```

Ecriture d'exemples de logique applicative efficaces

Les meilleures recommandations sur la façon d'écrire efficacement des logiques applicatives sont données sur les sujets suivants :

- **Métriques groupées**
  - Lors de l'évaluation du volume du système, considérez une métrique groupée comme le nombre d'éléments dans la métrique et souvenez-vous que tout est multiplié.
  - Le nouveau calcul d'un élément groupé engendre le nouveau calcul du groupe entier. Par conséquent, souvenez-vous en lors de la planification du groupement, de la manière dont les adaptateurs sont configurés mais également lorsque vous changez la structure des ressources.
  - Les mêmes événements de données brutes se retrouvant dans plusieurs éléments groupés ont un coût de performance élevé (commutation de contexte).
- **Variables globales**
  - Insérez les paramètres et valeurs d'attribut dans chaque endroit dans le code où ils sont nécessaires. Evitez de les garder dans une variable globale et plus particulièrement dans des métriques groupées (à défaut d'augmenter la taille des "états")
  - Evitez la logique traitant les grands plans. A la place, traitez chaque événement dans la méthode OnXXEvent
  - Supprimez des éléments des plans dès que possible. Par exemple lorsqu'un ticket est clôturé et non à la fin de la période
- **Modèles de conception**

  Le package de contenu prédéfini contient plusieurs modèles de conception destinés à des scénarios communs. L'utilisation de ces modèles de conception peut améliorer les performances
- **Fonctionnalité intégrée**

  L'ACE dispose d'une fonctionnalité intégrée et d'outils utilisés à différents fins :
  - Fonctionnalité de période d'application
    - NetTime
    - IsWithinTimeslot
  - Temps des événements
    - TimeOfLastEvent

        - TimeOfLastEventHandler
    - Objet de contexte
        - Contient de nombreuses méthodes sensibles à l'environnement
        - Utilisez ces méthodes et évitez "Safe ODBC"
- **Sorties de logique applicative**

  Gardez la structure dans T_SLALOM_OUTPUTS. Cela signifie qu'il peut être très utile de placer des champs logiques similaires dans le même champ physique si vous possédez plusieurs tables logiques ayant une structure similaire dans T_SLALOM_OUTPUTS. Cela permet une indexation facile pour une performance de rapports accrue
- **Réutilisabilité des événements**

  A utiliser lorsque :

    - Plusieurs métriques effectuent le premier calcul de phase, nécessaire au contrat et envoient des événements à une métrique récapitulative qui calcule le résultat (par ex. calcul financier, KPI de haut niveau)
    - Une métrique seule effectue une agrégation préliminaire avec des données brutes et envoie les événements à plusieurs autres métriques. Raisonnable lorsque la métrique n'envoie que peu d'événements par rapport à ce qu'elle reçoit, où lorsque celle-ci effectue de nombreux calculs qui serait autrement répétés plusieurs fois.

  A éviter lorsque :

    - Vous augmentez significativement le nombre de métriques
    - Vous implémentez plus de trois niveaux
    - La complexité de l'implémentation augmente et la maintenance devient moins aisée
- **Nouveaux calculs**
    - Evitez les nouveaux calculs massifs comme partie de l'opération normale du système
    - Les raisons pour de nouveaux calculs sont les suivantes :
        - Données brutes avec un horodatage passé
        - Singularité d'événement au passé qui change les données brutes
        - Corrections
        - Exceptions
        - Changements dans les modules de logique applicative

    - Changements dans un contrat
    - Evénements de réutilisabilité d'événement avec un horodatage passé
    - Changements dans la structure de ressources
  - Considérez l'utilisation de la fonctionnalité de verrouillage des données calculées.
- **Modules de logique applicative**
  - Les modules de logique applicative doivent être écrits de sorte qu'une fois complètement vérifiés, ceux-ci n'aient pas besoin d'être changés
  - La directive est la suivante : un module égale une logique générique
  - Les modules de logique applicative très spécifiques ne peuvent pas servir beaucoup de métriques, et ne promeuvent pas la réutilisabilité de code et de logique
  - Les modules de logique applicative trop génériques sont difficiles à maintenir. De plus, si un module de logique applicative implémente de nombreuses logiques complexes, un correctif dans un flux (utilisé par une partie des métriques) cause un nouveau calcul de toutes les métriques
- **Enregistrement**
  - Exécutez toute la filtration des événements en utilisant l'enregistrement. Laisser le filtre au code influence fortement les performances
  - Allez au plus simple
  - Pour le code n'appartenant pas à l'enregistrement, servez-vous des méthodes dispatcher.IsRunTimeMode et d'OnResourceStructureChange, particulièrement lors de nombreux changements de ressources
  - Evitez d'utiliser la méthode RegisterByEventType
  - Dans les modules de logique applicative, utilisez un formulaire générique (par contractant, service ou type de ressource), des paramètres, ou laissez l'enregistrement se faire au moyen de l'interface utilisateur (option conseillée pour la réutilisabilité d'événement)

Etude de cas 15 : Métriques groupées

La description d'une partie d'un logiciel peut généralement être divisée en deux parties, le QUOI et le COMMENT. QUOI signifiant la description de ce que la partie du code fait. Le COMMENT pour la façon dont il le fait. Si l'on se concentre souvent sur le QUOI, le COMMENT tend plutôt à être ignorer. La raison en est simple et dans beaucoup de cas, elle est justifiée. Vous réduisez ainsi la connexion entre vos composants et ne surchargez pas votre esprit avec des informations le plus souvent non pertinentes. Dans beaucoup de cas en revanche, ignorer le COMMENT implique des performances médiocres.

Cette étude de cas aborde la manière dont le moteur calcule des métriques groupées (réponse au COMMENT) et décrit le coût de performance impliqué pour certaines implémentations. Elle examine également plusieurs façons de réduire ce coût en changeant l'implémentation.

Que sont les métriques groupées ?

Les métriques groupées sont des métriques intégrant un certain groupe de ressources dans leur définition. Ce groupe est appelé groupe de la métrique et chacune des ressources dans ce groupe est appelée élément de groupe. Lors du calcul d'une métrique groupée, un calcul séparé est effectué pour chaque élément groupé. Les calculs de ces éléments sont identiques excepté pour :

- Context.ClusterItem : Valeur de la propriété Context.ClusterItem, égale à l'élément groupé en cours de calcul.
- Enregistrements par élément groupé : Lorsqu'une métrique est enregistrée en événements par élément groupé, chaque élément groupé reçoit des données brutes et des événements de réutilisabilité.

De quelle manière les métriques groupées sont-elles calculées ?

Ce qu'il importe de comprendre à propos du calcul d'une métrique groupée est que tous les éléments groupés sont calculés en parallèle. Nous n'entendons pas par là qu'ils sont calculés par fils différents, mais que lors du traitement des événements devant être gérés par les différents éléments groupés, ces événements sont traités de façon séquentielle, et pour chacun d'entre deux les éléments groupés appropriés sont appelés pour traiter l'événement. Par exemple, beaucoup d'événements doivent être traités par de nombreux éléments groupés. Il existe deux procédures à cette fin :

Exemple : option 1

```
For each cluster item C
     For each event E that should be handled by C
          Let C handle E
```

Exemple : option 2

```
For each event E
     For each cluster item C that should handle E
          Let C handle E
```

Le moteur gère les métriques groupées en utilisant l'option 2.

Autre point important : l'exécution du VBScript dans le PslWriter est accomplie par un composant appelé Contrôle de script. Il n'existe qu'une seule instance de ce composant pour chaque métrique et cette instance est réutilisée pour le calcul des divers éléments groupés. Vu que les éléments groupés sont calculés en parallèle et que le composant Script Control ne peut contenir les données que d'un seul élément groupé à chaque instant, nous devons fréquemment basculer les données dans le composant Script Control.

Pour expliquer cela, un code fictif plus détaillé effectuant le calcul est présenté ci-dessous.

```
1-     For each metric M
2-          Let X be the earliest event not yet handled by M
3-          Let T be the timestamp of the latest state before X
4-          Let L be the list of all events registered by M (all cluster items)
starting from timestamp T until the current time
5-          For each event E in L
6-               For each cluster item C that should handle E
7-                    Let C’ be the cluster item that is currently loaded into
the script control
8-                    Take the values of the global variables from the script
control and store them aside for C’
9-                    Take the values of the global variables stored aside for C
and load them into the script control
10-                    Handle event E
```

Tout le processus consistant à rechercher le moment d'un événement qui n'était pas encore pris en compte pour ensuite calculer depuis ce point est nommé Nouveau calcul. Le processus visant à remplacer les valeurs des variables globales (étapes 8 et 9 du code ci-dessus) est appelé Commutation de contexte.

Les deux problèmes principaux que vous pouvez facilement observer dans le code ci-dessus sont les suivants :

- Le nouveau calcul s'effectue pour l'ensemble des éléments groupés. Une fois que l'instant T (étape 3 dans le code ci-dessus) est trouvé, tous les éléments groupés effectuent un nouveau à partir de ce point. Cela signifie qu'à chaque fois qu'un élément groupé reçoit un nouvel événement, tous les éléments groupés sont recalculés.

- La commutation de contexte est très courante. Vous pouvez le voir facilement car la commutation de contexte (étapes 8 et 9 du code ci-dessus) est localisée dans la boucle intérieure.

Nouveau calcul de métriques groupées

- Description du problème

  Comme nous l'avons déjà expliqué, tous les éléments groupés d'une métrique groupée sont recalculés dans leur ensemble. Cela signifie que si nous avons une métrique groupée sur 1000 éléments groupés et que l'un d'entre eux nécessite un nouveau calcul de l'année précédente, suite à un nouvel événement reçu, alors les 1000 éléments groupés sont recalculés pour toute l'année précédente.

- Solutions possibles

  Les propositions de solutions suivantes peuvent aider à réduire ce problème, mais ils ne sont pas toujours applicables et chacun a ses inconvénients. Il importe de comprendre le problème et son coût approximatif et de comparer ce coût au coût des solutions suggérées.

  - Lorsque le nombre d'éléments groupés est faible, nous pouvons considérer de définir chaque élément comme métrique à part. L'inconvénient de cette approche est bien sûr le coût de maintenance qu'elle engendre, maintenir plusieurs métriques mais également ne pas pouvoir effectuer de rapport pour la métrique entière et parcourir un élément groupé spécifique
  - Lorsque le nombre d'éléments groupés est grand mais qu'un seul ou quelques uns d'entre eux sont fréquemment recalculés, nous pouvons étudier la possibilité de mettre ces éléments groupés dans une métrique à part et de laisser tous les autres éléments groupés dans l'autre métrique
  - Utilisez fréquemment un gel de calcul pour le contrat/métrique approprié afin que cette métrique n'ait jamais de nouveaux calculs trop longs
  - Procédez à quelques changements dans les adaptateurs et sources de données pour ne plus avoir de long nouveaux calculs (c.-à-d. ne pas envoyer d'événements dont l'horodatage date de plus d'un mois)

Commutation de contexte

- Description du problème

Comme précédemment expliqué, la commutation de contexte se fait dans la boucle la plus intérieure. En d'autres termes, pour chaque événement devant être géré par chaque élément groupé. Lorsqu'une métrique reçoit de nombreux événements et que chacun est géré par beaucoup d'éléments groupés, cette quantité peut être très élevée. Ajoutez à cela que l'opération de commutation de contexte est assez coûteuse (par rapport au traitement de l'événement lui-même dans la logique applicative) et un problème se pose.

Le coût de l'opération de commutation est proportionnel à la taille des données qui sont "commutées". Les données que nous basculons durant la commutation de contexte sont les valeurs de toutes les variables globales dans notre logique applicative (également appelées "l'état"). Ainsi, plus vous avez de variables globales et plus leur taille est grande, plus la commutation de contexte est coûteuse.

Il n'est pas recommandé d'utiliser des plans de logique applicative dans des métriques groupées, particulièrement si la taille de ces plans peut être grande.

- Solutions possibles
  - Réduisez le temps de chaque commutateur de contexte

    L'idée est de réduire la taille de l'état (variables globales). Vous pouvez adopter cette approche en réécrivant la logique applicative pour qu'elle ne contienne aucun plan. Bien sûr ce n'est pas toujours possible, mais néanmoins recommandé lorsque c'est le cas.
  - Réduire la quantité des commutateurs de contexte

    Lorsque le groupement est petit, il est possible de créer une métrique à part pour chaque élément groupé.

    Evitez les métriques groupées qui comprennent de nombreux éléments groupés s'enregistrant auprès des mêmes événements. L'idée ici est la suivante :

    Si chaque événement est géré par un seul élément groupé, alors le montant de la "commutation de contexte" est proportionnelle au nombre d'événements

    Si chaque événement est géré par tous les éléments groupés, le montant de la "commutation de contexte" est proportionnel au nombre d'événements fois le nombre d'éléments groupés.

Créez une métrique non groupée qui calculera le résultat pour tous les éléments groupés d'origine (maintenant simples ressources et plus éléments groupés). Faites en sorte que cette métrique envoie le résultat de chaque élément groupé en tant qu'événement. Créez une autre métrique groupée pour recevoir les événements de la première métrique, puis rapportez les valeurs reçues dans ces événements en tant que résultats. L'idée ici est que la grande quantité des événements de données brutes sera traitée par une métrique non groupée, tandis que la métrique groupée traitera un seul événement par période par élément groupé.

Etude de cas 16 : Modèles de conception de logique applicative

Il existe plusieurs "Modèles de conception" que vous pouvez utiliser dans un grand nombre de logiques applicatives. Ces modèles de conception sont testés, leur utilisation (lorsqu'elle est possible) peut faire gagner un temps précieux et la plupart du temps créer une logique applicative plus efficace. Cette étude de cas se concentre sur l'un de ces modèles de conception.

Mettre à jour les modèles de conception de compteurs

Ce modèle de conception conçu pour mesurer le temps entre certains événements est utile dans presque toutes les logiques applicatives. De telles logiques applicatives peuvent servir à mesurer le temps de disponibilité, le temps d'indisponibilité, la durée moyenne entre deux pannes, la durée moyenne nécessaire pour restaurer, le temps de réponse moyen, la durée moyenne de résolution, le pourcentage de composants ayant une disponibilité inférieure à X, le nombre de cas non résolus à temps, etc.

Le point commun à toutes ces logiques applicatives est que leur résultat dépend de l'horodatage des divers événements qu'elles reçoivent.

Un principe de base pour décider si votre logique applicative peut bénéficier de ce modèle de conception serait : si votre logique applicative dépend de l'horodatage des divers événements qu'elle reçoit, il doit fort probablement utiliser ce modèle de conception.

Squelette de ce modèle de conception

Vous pouvez fractionner le code d'une logique applicative qui utilise ce modèle en deux parts : un cadre et une implémentation. Le cadre contient un code qui dans la plupart des cas est fixé et ne change pas pour les diverses logiques applicatives. Cette partie est la même pour le calcul de disponibilité et le nombre de tickets non résolus à temps. L'implémentation contient un code qui est spécifique à chaque logique applicative.

Il est recommandé de mettre ces deux parties du code dans des modules de logique applicative distincts et de réutiliser le module de la structure dans des métriques différentes.

Voici le code de la structure :

```
Dim g_PrevEventTimestamp

Sub OnLoad(time)
     g_PrevEventTimestamp = time
     InitializeStatusVariables
End sub

Sub OnRegistration(Dispatcher)
     'S'il y a une copie distincte de variables de statut
     'Pour chaque ressource enregistrée dépendre des ressources enregistrées
     'Vous devriez définir des valeurs initiales sur les
     'Variables de statut des ressources nouvellement ajoutées ici
End sub

Sub OnPeriodStart(time)
     InitializeCounters
End sub

Sub OnPeriodEnd(time, completePeriod)
     HandleEvent (time)
End sub

Sub OnTimeslotEnter(time)
     HandleEvent (time)
End sub

Sub OnTimeslotExit(time)
     HandleEvent (time)
End sub


Function Result()
     Result = CalculateResult()
End Function
```

```
Sub HandleEvent(Time)
     Dim diff
     diff = TimeDiff( "s",Time,g_PrevEventTimestamp)
     UpdateCounters(diff)
     g_PrevEventTimestamp = Time
End sub
```

Voici un squelette du module d'implémentation :

```
'Définissez vos variables de statut ici. Ce peut être une
'Simple variable globale ou de nombreuses variables globales complexes
'Selon la logique applicative
Dim g_StatusVar_1, g_StatusVar_2, ... ,g_StatusVar_n

'Définissez vos compteurs ici.
'Cela peut être une simple variable globale ou de nombreuses
'Variables globales complexes selon la logique applicative
Dim g_Counter_1, g_Counter_2, ... , g_Counter_n

Sub InitializeStatusVariables ()
     'Définir des valeurs initiales sur les diverses variables de statut
End sub

Sub InitializeCounters ()
     'Définir des valeurs initiales sur les divers compteurs
     g_Counter_1 = 0
     g_Counter_2 = 0
     '…
     g_Counter_n = 0
End sub

Function CalculateResult ()
     'Calculer le résultat. Le résultat doit varier selon
     'Les valeurs des compteurs. Il ne doit pas varier selon
     'La valeur des variables de statut. Il ne doit pas
     'Modifier les valeurs des compteurs ou les variables
     'de statut
End Function

Sub UpdateStatus(method, eventDetails)
     'Mettre à jour la valeur des variables de statut selon
     'Les paramètres (et probablement selon l'ancienne valeur des
     'Variables de statut)
End sub

Sub UpdateCounters(diff)
     'Mettre à jour les valeurs des compteurs selon leur
```

```
     'Valeur précédente, selon la valeur des variables de statut
     'Et selon la valeur du paramètre de diff.
     'Dans beaucoup de cas, ce calcul varie également selon la
     'Valeur de Context.IsWithinTimeslot
End sub

Sub OnEvent_1(eventDetails)
     HandleEvent (eventDetails.Time)
     UpdateStatus("OnEvent_1",eventDetails)
End sub

Sub OnEvent_2(eventDetails)
     HandleEvent (eventDetails.Time)
     UpdateStatus("OnEvent_2",eventDetails)
End sub

'...

Sub OnEvent_n(eventDetails)
     HandleEvent (eventDetails.Time)
     UpdateStatus("OnEvent_n",eventDetails)
End sub
```

Afin de mieux expliquer ce modèle de conception, voici un exemple d'implémentation de ce modèle permettant de calculer la disponibilité. Cet exemple suppose qu'il existe des événements pour HAUT et BAS gérés par deux gestionnaires d'événements distincts dans la logique applicative. La disponibilité est définie comme le pourcentage de durée pendant laquelle le système affiche une durée supérieure à la durée totale de la période d'application. Il est supposé que le statut du système est le statut du dernier événement reçu (HAUT ou BAS).

Voici le code d'implémentation (le code du cadre reste inchangé) :

```
'Variable de statut
Dim g_Status
'Compteurs.
Dim g_UpTime, g_TotalTime

Sub InitializeStatusVariables ()
     G_Status = “UP”
End sub

Sub InitializeCounters ()
     g_UpTime = 0
     g_TotalTime = 0
End sub

Function CalculateResult ()
```

```
     If g_TotalTime = 0 Then
          CalculateResult = Null
     Else
          CalculateResult = g_UpTime/g_TotalTime*100
     End If
End Function

Sub UpdateStatus(method, eventDetails)
     If method = “OnUP” Then
          G_Status = “UP”
     Else
          G_Status = “DOWN”
     End If
End sub

Sub UpdateCounters(diff)
     If Context.IsWithinTimeslot Then
          G_TotalTime = g_TotalTime + diff
          If g_Status = “UP” Then
               G_UpTime = g_UpTime + diff
          End If
     End If
End sub

Sub OnUp(eventDetails)
     HandleEvent (eventDetails.Time)
     UpdateStatus(“OnUp”,eventDetails)
End sub

Sub OnDown(eventDetails)
     HandleEvent (eventDetails.Time)
     UpdateStatus(“OnDown”,eventDetails)
End sub
```

Ce modèle présente plusieurs variations. Une des variations les plus communes est lorsqu'un compteur de temps distinct doit être maintenu pour des entités différentes. Par exemple, lorsque nous évaluons le temps de résolution, un compteur distinct doit être maintenu pour chaque ticket ouvert. Dans ce cas lors de la gestion d'un événement ne concernant qu'un ticket, il est plus judicieux de mettre à jour uniquement le compteur de ce ticket. Lorsqu'un événement commun est géré (comme l'OnPeriodEnd ou l'OnTimeslotEnter), les compteurs de tous les tickets doivent être mis à jour.

Remarque : Cette variation du modèle nécessite le maintien d'une copie distincte de la variable globale de g_PrevEventTimestamp de chaque ticket.

Vous pouvez consulter quelques exemples appropriés de l'utilisation de ce modèle dans notre contenu prédéfini. Rappelez-vous cependant que ce modèle est utilisé un peu différemment dans le contenu prédéfini et que la distinction entre la structure et l'implémentation n'est pas si évidente dans ce cas là.

Etude de cas 17 : Fonctionnalité intégrée

Le programme ACE comporte une fonctionnalité intégrée et des outils pour divers buts. Il est préférable d'utiliser cette fonctionnalité intégrée plutôt que de l'écrire en VBS. Puisque VBS est un langage interprété, reproduire cette fonctionnalité en VBS en altère la performance.

Voici une liste des fonctions intégrées et de la façon appropriée de les utiliser :

IsWithinTimeslot

Voici la fonction intégrée la plus simple. Elle a pour but de permettre à la logique applicative de définir si le système se trouve actuellement dans une période d'application. Cela épargne la nécessité de gérer une variable dans les fonctions entrée et sortie de la période d'application pour effectuer la même opération. Par exemple, au lieu d'exécuter le code suivant :

```
Dim amIWithinATimeslot
Sub OnTimeslotEnter(time)
     amIWithinATimeslot = 1
End Sub
Sub OnTimeslotExit(time)
     amIWithinATimeslot = 0
End Sub
Sub OnEvent(eventDetails)
     If amIWithinATimeslot = 1 Then
           count = count + 1
```

```
     End if
End Sub
```

A la place, vous pouvez exécuter ce code beaucoup plus simple :

```
Sub OnEvent(eventDetails)
     If context.IsWithinTimeslot Then
          count = count + 1
     End if
End Sub
```

Si vous souhaitiez utiliser ou sauvegarder des informations à propos de l'horodatage des fonctions entrée et sortie de la période d'application, cette fonctionnalité ne couvrirait pas vos besoins. Mais normalement, cela n'est pas nécessaire. Ce code suffit.

TimeOfLastEvent

Cette fonction vous donne l'horodatage des dernières données brutes ou de l'événement de données intermédiaires géré. Cela veut dire que vous ne devez pas sauvegarder ces informations dans le gestionnaire d'événements, puisque vous pouvez y accéder directement à travers cette fonction. Par exemple :

```
Function result
     Dim LastEventTimestamp
     LastEventTimestamp = Context.TimeOfLastEvent
End function
```

TimeOfLastEventHandler

Cette fonction permet de retourner à l'horodatage du dernier gestionnaire d'événements appelé par ACE. Elle inclut non seulement des gestionnaires d'événements de données brutes et intermédiaires, mais également tout événement système qui aurait aussi été appelé. Elle est particulièrement utile pour des gestionnaires d'événements qui ne reçoivent pas le temps de la fonction de résultat, par exemple. Par exemple :

```
Function result
     Dim LastEventHandlerTimestamp
     LastEventHandlerTimestamp= Context.TimeOfLastEventHandler
End function
```

NetTime

Cette fonction vous permet de spécifier deux horodatages et de recevoir le temps net (en secondes) pendant lequel le système s'est trouvé dans la période d'application de la règle actuelle, intervalle entre ces deux horodatages. Il s'agit particulièrement d'une fonctionnalité peu pratique qui ne doit pas être implémentée dans VBS. L'implémenter dans VBS signifie conserver une liste de toutes les entrées et sorties de la période d'application ou calculer directement la différence entre chaque temps de sortie de la période d'application, afin d'évaluer la période de temps entre eux. Dans des cas extrêmes, cela pourrait se produire un grand nombre de fois et ce ne serait pas avantageux pour la vitesse de calcul. La fonction interne agit de la même façon après une optimisation significative, et avec plus d'efficacité. Par exemple :

```
Function result
     Dim MyNetTime
     MyNetTime = Tools.NetTime(MyBeginTimestamp, MyEndTimestamp)
End function
```

L'objet du contexte

L'objet du contexte a plusieurs paramètres qui fournissent des informations sur :

- la métrique actuelle,
- le contrat où il est impliqué,
- l'état de calcul actuel,
- le domaine de service, la catégorie de domaine et leurs valeurs correspondantes (p. ex. seuil).
- Informations sur le groupement : Le groupe en général et l'élément de groupe spécifique en cours de gestion.
- Fonctions qui renvoient des listes de ressources en fonction des besoins de l'utilisateur.
- Fonctions qui vous permettent de convertir des noms de ressource en ID de ressource et vice versa.

L'accès direct à ces informations depuis la base de données à l'aide de Safe ODBC est extrêmement inefficace et n'a pas d'importance puisque les informations sont directement accessibles à partir de l'objet de contexte. Si possible, l'utilisation continue de la fonctionnalité intégrée constitue un moyen de recevoir des informations.

Etude de cas 18 : Enregistrement

La logique applicative est souvent écrite de façon à conserver un plan de la structure de ressource de la métrique à utiliser pendant les calculs. Puisque la structure de la ressource change avec le temps, cette logique applicative doit mettre à jour la structure du plan lorsque la structure de ressource change.

La méthode OnRegistration est appelée lorsque la structure de ressource change, puisqu'elle est chargée de gérer le comportement de moteur qui a un lien avec les changements observés dans les enregistrements et le groupement en raison des changements de structure de ressource. Le fait que cette méthode est appelée pour chaque changement de structure de ressource constitue un atout pour mettre à jour le plan mentionné ci-dessus. Toutefois, le remplissage du plan n'est pas utile à l'enregistrement. Cela signifie que le remplissage du plan compromet la performance de la fonction OnRegistration. Ce n'est pas important pendant le temps d'exécution, puisqu'il ne se produit pas très souvent. Toutefois, la méthode OnRegistration est également appelée pendant le processus de traitement de l'infrastructure du moteur, pendant lequel le système définit si les changements de structure de ressource sont utiles à l'enregistrement de chaque métrique spécifique dont l'instance est responsable. Pendant ce processus, la méthode OnRegistration est appelée pour tous les changements de la structure de ressource, même si le changement de structure n'est pas utile à la métrique actuelle. Cela signifie que la méthode peut être appelée un grand nombre de fois par métrique.

Si cette logique est implémentée dans la méthode OnRegistration, une petite dégradation de performance pendant le temps d'exécution pourrait devenir une dégradation de performance très importante pendant le traitement de l'infrastructure.

Pour résoudre ce problème, il existe deux manières de remplir des plans ou toute autre initialisation qui doit être exécutée lorsqu'il y a changement de la structure de ressource, non utile à l'enregistrement :

Utilisation de la propriété IsRunTimeMode dans l'objet distributeur.

Cette propriété permet à l'utilisateur de déterminer si l'exécution actuelle est une exécution de calcul ou non, et d'enfermer la logique qui n'est pas utile à l'enregistrement dans un énoncé "if" qui veillera à ce que la logique s'exécute uniquement pendant le temps d'exécution.

Dans l'exemple ci-dessous, la partie marquée en bleu est la partie de la logique applicative utile à l'enregistrement et qui doit toujours être exécutée. La partie marquée en vert n'est pas utile à l'enregistrement et peut être encadrée dans le nouvel énoncé "if".

```
Sub OnRegistration (dispatcher)
     Dim MyResource
     MyResource = Context.ClusterItem
     Dispatcher.RegisterByResource “OnEvent”, “My Event Type”, MyResource
     Dim ThisResourceMap
     Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
     Dim resource
     Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add resource
     Next
End sub
```

Vous pouvez améliorer ce code en le modifiant de la façon suivante :

```
Sub OnRegistration (dispatcher)
     Dim MyResource
     MyResource = Context.ClusterItem
     Dispatcher.RegisterByResource “OnEvent”, “My Event Type”, MyResource
     If Dispatcher.IsRunTimeMode Then
          Dim ThisResourceMap
          Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
          Dim resource
          ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
          For Each resource In ThisResourceMap
               GlobalResourceVector.Add resource
          Next
     End If
End sub
```

Utiliser la méthode OnResourceStructureChanged.

Cette méthode s'exécute juste après la méthode OnRegistration et produit donc la même fonctionnalité que la méthodologie d'origine, mais elle s'exécute uniquement pendant le temps d'exécution. Cette méthode n'est pas appelée pendant le traitement d'infrastructure et la performance n'est donc pas endommagée.

Dans l'exemple ci-dessous, la partie marquée en bleu est la partie de la logique applicative utile à l'enregistrement et qui doit toujours être exécutée. La partie marquée en vert n'est pas utile à l'enregistrement et peut être placée dans la nouvelle fonction.

```
Sub OnRegistration (dispatcher)
     Dim MyResource
     MyResource = Context.ClusterItem
     Dispatcher.RegisterByResource “OnEvent”, “My Event Type”, MyResource
     Dim ThisResourceMap
     Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
     Dim resource
```

```
     Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add resource
     Next
End sub
```

Vous pouvez améliorer ce code en le modifiant de la façon suivante :

```
Sub OnRegistration (dispatcher)
     Dim MyResource
     MyResource = Context.ClusterItem
     Dispatcher.RegisterByResource “OnEvent”, “My Event Type”, MyResource
End sub

Sub OnResourceStructureChanged(time)
     Dim ThisResourceMap
     Set GlobalResourceVector= CreateObject("SlalomVector.Vector")
     Dim resource
     Set ThisResourceMap = Context.ResourcesOfResourceGroup(Context.ClusterItem)
     For Each resource In ThisResourceMap
          GlobalResourceVector.Add resource
     Next
End sub
```

Etude de cas 19 : Assistant de configuration d'adaptateur pour une source de données basée sur un fichier

Cette étude de cas examine les bonnes pratiques d'intégration à l'aide une source de données basée sur un fichier. L'exemple de scénario traite un fichier de données CSV produit par le système source. Pour la plupart des intégrations basées sur un fichier, CA recommande, conformément à certaines directives de base, de réduire autant que possible le risque à l'intégration. Ces directives sont les suivantes :

- Lorsque l'option est disponible, nous devons exiger que les données soient transmises dans le système de fichier du serveur d'application CA Business Service Insight. Le mécanisme de transmission ne dépend ainsi pas de l'adaptateur qui essaie d'obtenir des données d'un magasin distant (réduit les problèmes d'autorisation des comptes d'utilisateurs et les problèmes de synchronisation).
- La convention d'attribution de nom à un fichier est importante d'autant que l'adaptateur classe les fichiers selon l'attribution alphanumérique de noms. Si nous pouvons la contrôler, CA recommande de demander deux parties :
  - Une convention d'attribution de nom raisonnable basée sur le contenu du fichier source (en particulier si plusieurs fichiers proviennent de la source)
  - Un horodatage à ordre inverse pour veiller à ce que les fichiers soient classés avec le fichier le plus récent en dernier. (c.-à-d., *<file_name>_aaaammjj_hhmiss.csv*). La profondeur de l'horodatage utilisé dépendrait de la fréquence des données transmises

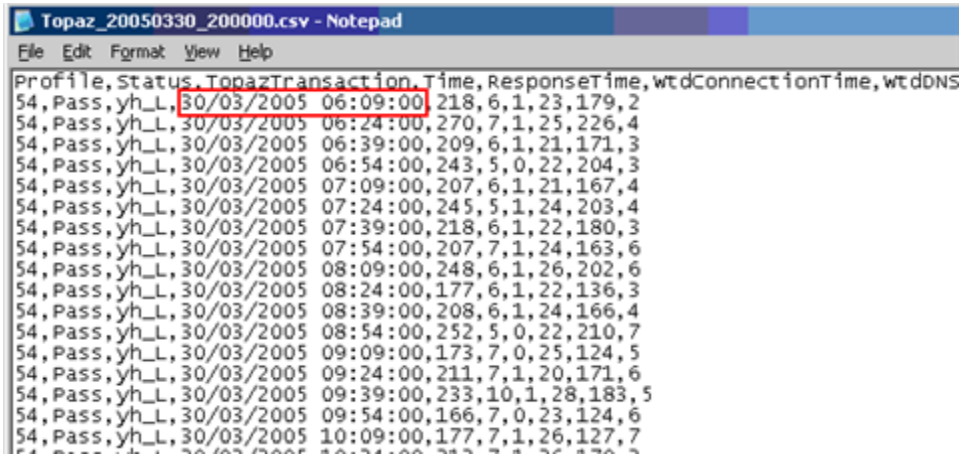
Dans ce scénario, les fichiers source proviennent d'une source de données Topaz (actuellement connue sous le nom de Moniteur Global de Mercure, propriété de HP). Elle a été créée à l'aide d'une API du produit afin de contenir les fichiers obligatoires des KPI spécifiques obligatoires. Les fichiers sont directement distribués au serveur d'application CA Business Service Insight par un processus automatisé externe. Les fichiers source sont nommés : *Topaz_aaaammjj_hhmiss.csv*.

Remarque : L'horodatage du fichier est l'heure à laquelle il a été créé, c'est pourquoi toutes les entrées du fichier se rapportent à cette heure.

Vous pouvez consulter un échantillon des données dans le fichier ci-dessous.

	A	B	C	D	E	F	G	H	I	J	K
1	Profile	Status	TopazTransaction	Time	ResponseTime	WtdConnectionTime	WtdDNSTime	WtdSSLTime	WtdNetworkTime	WtdDownloadTime	
2	54	Pass	yh_L	30/03/2005 06:09	218	6	1	23	179	2	
3	54	Pass	yh_L	30/03/2005 06:24	270	7	1	25	226	4	
4	54	Pass	yh_L	30/03/2005 06:39	209	6	1	21	171	3	
5	54	Pass	yh_L	30/03/2005 06:54	243	5	0	22	204	3	
6	54	Pass	yh_L	30/03/2005 07:09	207	6	1	21	167	4	
7	54	Pass	yh_L	30/03/2005 07:24	245	5	1	24	203	4	
8	54	Pass	yh_L	30/03/2005 07:39	218	6	1	22	180	3	
9	54	Pass	yh_L	30/03/2005 07:54	207	7	1	24	163	6	
10	54	Pass	yh_L	30/03/2005 08:09	248	6	1	26	202	6	
11	54	Pass	yh_L	30/03/2005 08:24	177	6	1	22	136	3	
12	54	Pass	yh_L	30/03/2005 08:39	208	6	1	24	166	4	
13	54	Pass	yh_L	30/03/2005 08:54	252	5	0	22	210	7	
14	54	Pass	yh_L	30/03/2005 09:09	173	7	0	25	124	5	
15	54	Pass	yh_L	30/03/2005 09:24	211	7	1	20	171	6	

Remarque : CA recommande de vérifier des fichiers CSV dans le Bloc-notes (plutôt que dans Excel uniquement) afin de veiller à ce que le format de la date soit comme prévu. Excel a tendance à formater les dates selon des paramètres régionaux de l'ordinateur et peut ne pas faire correspondre le véritable format vu par l'adaptateur.



Avant de commencer la création d'un adaptateur, vous devez également avoir effectué toute l'analyse nécessaire et les recherches dans la source de données et les KPI correspondants afin de veiller à ce que les éléments suivants soient connus :

- les champs exigés par la logique applicative ;
- le format des dates du fichier ;
- le fuseau horaire des valeurs de date/heure du fichier source (ces fuseaux horaires doivent être créés dans le système avant de commencer la création d'un adaptateur) ;
- l'existence ou non de champs de date pouvant avoir des entrées vides/NULL ;
- le comportement de la source de données (tous les enregistrements sont-ils ajoutés ou certains enregistrements sont-ils une mise à jour d'un événement précédent) ;
- les zooms avant obligatoires pour les KPI (cela peut influer sur votre choix de ressource) ;
- les types d'événement nécessaires pour un filtrage approprié des événements contenus dans la logique applicative.

Une fois que ces points sont connus, vous pouvez commencer la création.

Maintenant, nous créons l'adaptateur sur la base de ce scénario, à l'aide de l'assistant de configuration d'adaptateur.

Dans notre scénario, TopazTransaction représente la ressource, Profil, le type d'événement et le champ Heure, l'horodatage. Nous ramenons également les champs Status, ResponseTime et WtdDownloadTime dans la structure d'événement, afin de les utiliser avec la logique applicative.

Création d'adaptateur

Tout d'abord, assurez-vous que le système est prêt à créer le nouvel adaptateur et à le déployer correctement sur le serveur en vérifiant que les services suivants sont démarrés sur le serveur d'application.

- Service de déploiement d'adaptateur
- Service d'écouteur d'adaptateur

Ensuite, ouvrez la page Adaptateurs et créez un nouvel adaptateur. Vous devez choisir l'option Ajouter > Adaptateur de fichier texte, à partir de la page Adaptateurs.

Etape Général

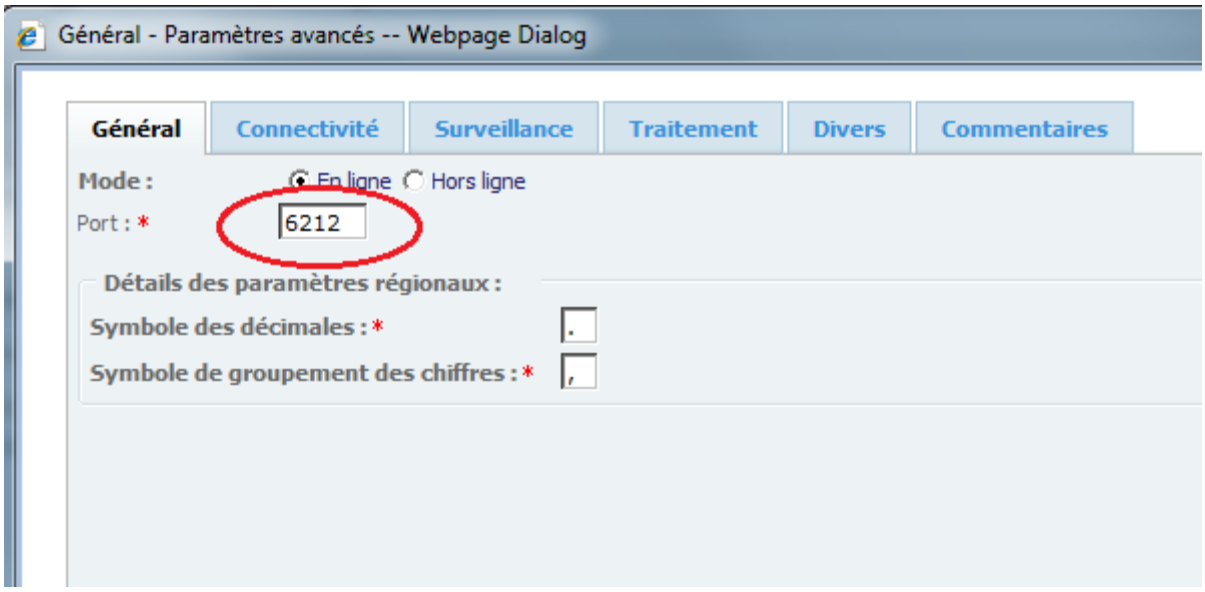
Dans l'étape Général de l'Assistant de configuration d'adaptateur, renseignez les champs suivants :

- *Nom* : Attribuez un nom convenable à l'adaptateur
- *Adresse de l'adaptateur* : LOCALHOST est l'option par défaut (pour le déploiement du serveur d'application), mais vous pouvez saisir d'autres adresses en utilisant le bouton en même temps, si nécessaire.
- *Format de date et d'heure* : format horaire par défaut utilisé dans les champs date/heure de la source de données. Un choix correct de ces éléments permet une détection automatique ultérieure de ces champs dans le processus de l'assistant. Vous pouvez saisir de nouveaux formats de date et d'heure dans ce champ si nécessaire en utilisant le bouton situé en regard de ce champ.
- *Fuseau horaire* : fuseau horaire selon lequel les enregistrements de données sont effectués. Ce point est nécessaire pour que l'adaptateur puisse ramener correctement le champ event_timestamp (et d'autres champs date/heure) à UTC pour une modification interne correcte de l'heure. Cela aurait dû être précédemment saisi selon la liste de contrôle de la source de données dans la section précédente.

Remarques :

Il existe également un bouton Avancé sur cette page qui permet d'accéder à un certain nombre de paramètres de configuration pour l'adaptateur. Vous pouvez conserver les valeurs par défaut de la plupart de ces paramètres, sauf si une modification s'avère nécessaire.

Le Port d'adaptateur est automatiquement affecté à l'adaptateur à compter de la période dès CA Business Service Insight, mais peut être remplacé dans ce champ si nécessaire. Parmi les autres paramètres importants pouvant être modifiés, citons les suivants : les paramètres régionaux, le mode en ligne/hors ligne, les détails de connexion, les options de surveillance et de journalisation, les paramètres Exécuter une fois/toujours, les limites d'erreur, les noms de fichier et les commentaires.



L'examen de chacun de ces paramètres n'entre pas dans le cadre de cette étude de cas, mais est disponible à la section Spécifications de configuration de l'adaptateur (page 377).

Cliquez sur Suivant pour passer à l'étape suivante de l'assistant. L'étape suivante donne accès à l'interface de source de données de l'adaptateur.

Etape Interface de source de données

Dans l'étape Interface de source de données de l'Assistant de configuration d'adaptateur, renseignez les champs suivants :

- *Nom de la source de données* : nom de ce fichier source précis (un adaptateur peut comporter plusieurs fichiers source)
- *Chemin du fichier* : le chemin d'accès sur le serveur d'application (ou autre serveur) que contient les fichiers de données source. Pour un serveur autre que le serveur d'application, utilisez une notation UNC (c.-à-d., //server01/sourcefolder)
- *Modèle de nom* : l'utiliser avec un caractère générique pour filtrer les fichiers situés au chemin du fichier chargés par l'adaptateur.

L'onglet Définition de l'analyse permet également de définir la structure du fichier en cours d'importation. Vous pouvez utiliser les champs comme suit :

- *Titre* : case à cocher de valeurs booléennes pour déterminer s'il existe ou non une rangée titre dans le fichier de données (c.-à-d., la première rangée du CSV est un nom de titre suivi par les valeurs inscrites sur les rangées suivantes)
- *Délimiteurs* : spécifiez le délimiteur du fichier qui sépare chaque champ.

Remarque : Il existe également deux autres boutons Avancé ici qui fournissent d'autres options de configuration. Un bouton se trouve sous l'onglet Détails, l'autre sous l'onglet Définition de l'analyse. Le bouton Avancé de l'onglet Détails donne accès aux paramètres suivants :

Source de données active : case à cocher booléenne qui vous permet d'activer ou de désactiver cette section source particulière de l'adaptateur

Après traitement : vous permet de déterminer si le fichier est supprimé ou conservé après le traitement

Nom de fichier initial : définit le nom du fichier initial sur lequel doit commencer le traitement (au cas où vous ne voulez pas charger tous les fichiers dans un répertoire particulier)

Contrôler les données tous les : définit l'intervalle de temps entre la vérification de la présence des nouveaux fichiers lorsque l'adaptateur est configuré pour une exécution continue

Le bouton Avancé de l'onglet Définition d'analyse permet de définir les options d'arrêt d'enregistrement comme les enregistrements à lignes multiples, etc.

Une fois que les détails et les définitions d'analyse sont définis, il est possible de charger un échantillon du fichier source dans l'assistant pour tester les options de configuration et prévisualiser le contenu du fichier.

En cliquant sur le bouton Parcourir, vous pouvez charger un exemple de fichier dans la fenêtre d'aperçu ci-dessous. Ouvrez l'exemple de fichier et appuyez sur le bouton Fichier test. Cette étape est facultative.

Remarque : Ouvrez la fonction Parcourir à partir de l'ordinateur local sur lequel vous travaillez. S'il ne s'agit pas du serveur d'application, vous devez avoir une copie des fichiers source qui sont sur le serveur pour que cela fonctionne. Vous ne pouvez pas parcourir directement le serveur à l'aide de cette fonction.

Une fois que le fichier est chargé, vous devez en apercevoir le contenu dans la fenêtre d'aperçu, tel qu'illustré ci-dessous.

Remarque : Le nom Champ est affiché dans l'en-tête, ainsi que le type de champ détecté (nombre entier, chaîne, heure, etc). Avant de continuer, vous devez vérifier que ceux-ci ont été correctement détectés et conformément à vos exigences. Bien entendu, vous devez veiller à ce que :

- Votre horodatage ait été détecté en tant qu'heure - cela se fait selon le format de temps spécifié à la première étape de l'assistant
- Votre ressource soit détectée en tant que chaîne

Lorsque vous êtes satisfait de l'aperçu du fichier, cliquez sur Suivant. L'étape Mappage s'affiche.

Etape Mappage

L'étape suivante (et finale) de l'assistant de configuration d'adaptateur accomplit les tâches de conversion et vous permet de convertir les champs de saisie de la source de données en champs de sortie, ce qui crée l'Evénement CA Business Service Insight . Il existe deux options pour continuer à partir de cette étape, selon que le Type d'événement est déjà créé dans le système ou non. Il existe également un certain nombre d'autres options que vous pouvez sélectionner afin de configurer et de veiller au respect des normes d'attribution de noms. Toutefois, la plupart d'entre elles sont facultatives, de manière à faciliter le processus et à réduire les étapes nécessaires. Les étapes obligatoires sont notées ci-dessous.

Les étapes de configuration de l'étape Mappage sont les suivantes (étapes facultatives incluses) :

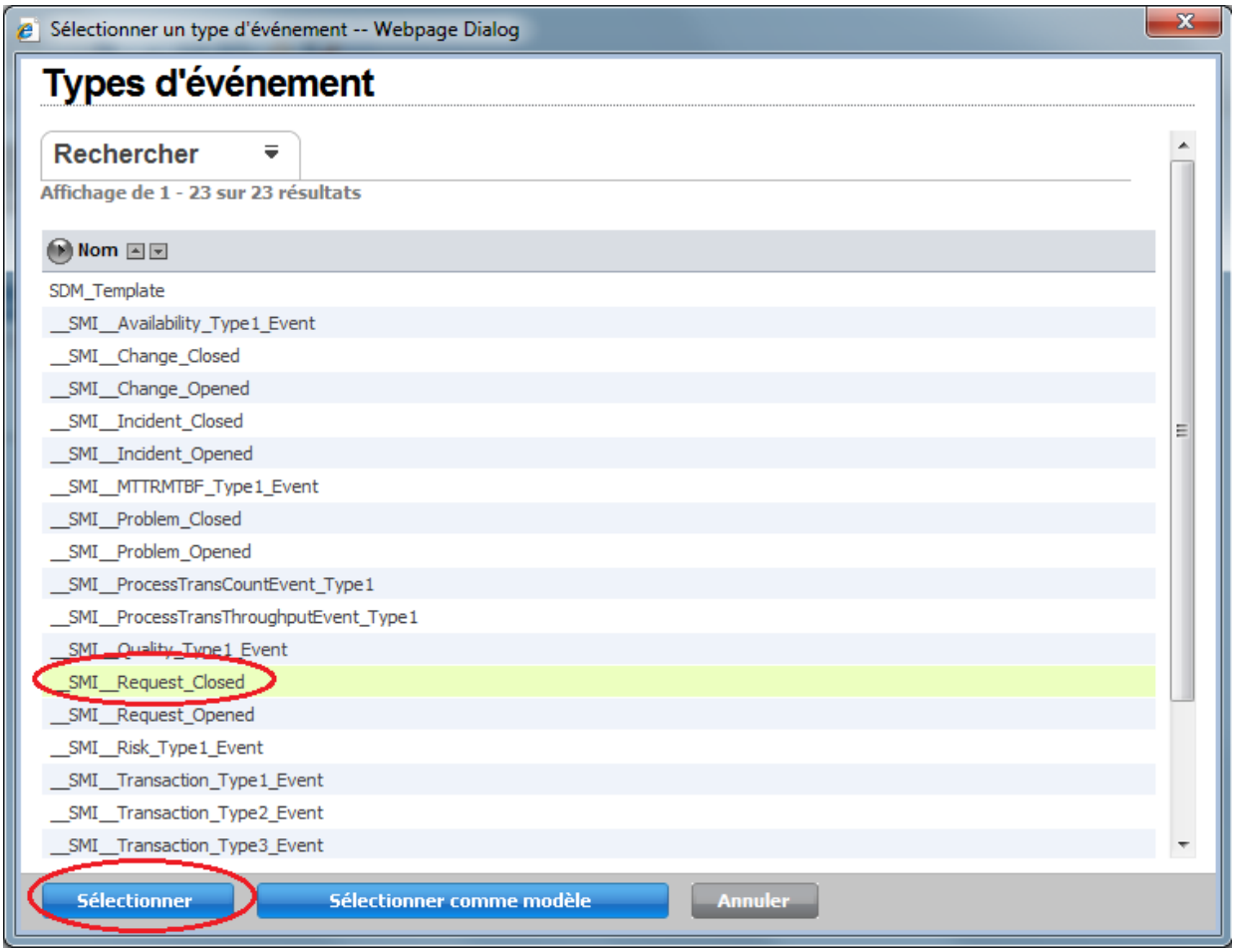
1. Fournir un nom au format d'entrée (utile pour des adaptateurs avec plusieurs entrées)
2. Ajouter tout autre champ nécessaire (comme Valeurs constantes, Source de données, Titre, Nom de fichier ou Types de champ composé)
3. Créer tout critère de validation d'entrée obligatoire
4. Sélectionner un type d'événement existant ou créer un nouveau type d'événement. (obligatoire)
5. Convertir les champs d'entrée en sortie (obligatoire)
6. Attribuer un nom au format de sortie
7. Mapper les éléments ResourceId, Horodatage et Type d'événement

8. Créer tout critère de validation de sortie nécessaire
9. Spécifier le paramètre "OnDuplication" des événements

Lorsque vous choisissez de créer un nouveau type d'événement, une nouvelle fenêtre apparaît et est pré-remplie, selon le format d'entrée de la fenêtre principale. Vous devez encore entrer le nom du type d'événement et affecter également un type de ressource au type d'événement.

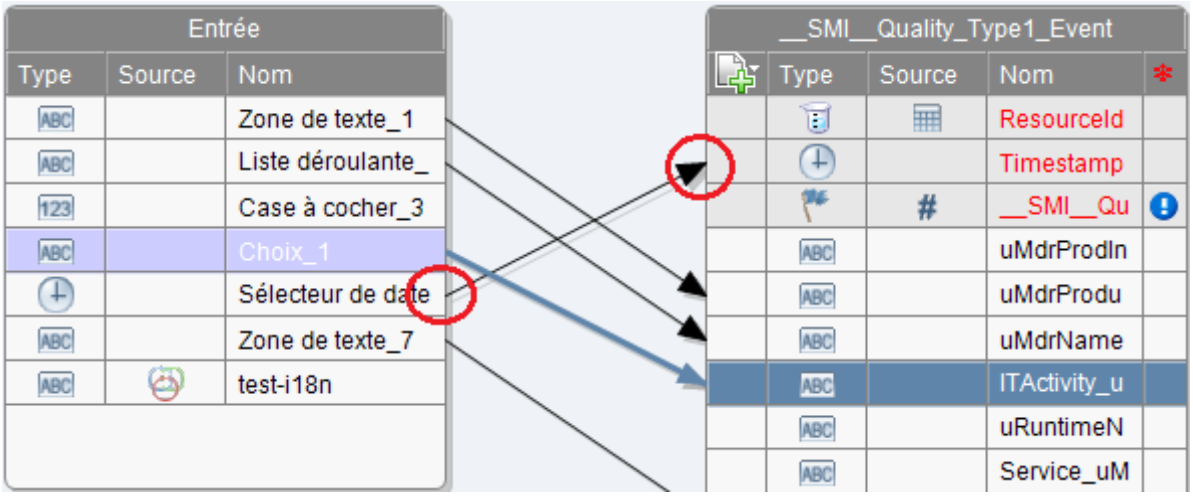
Une fois l'opération terminée, vous pouvez enregistrer et fermer ; le mappage s'achève automatiquement.

Si vous choisissez l'option Sélectionner un type d'événement, il apparaît une liste contenant les types d'événements existants dans le système parmi lesquels vous devez choisir. Toutefois, lorsque vous continuez, seuls les champs de l'entrée avec un nom correspondant et un type de données provenant du type d'événement sont automatiquement liés. Les autres champs doivent être mappés manuellement.

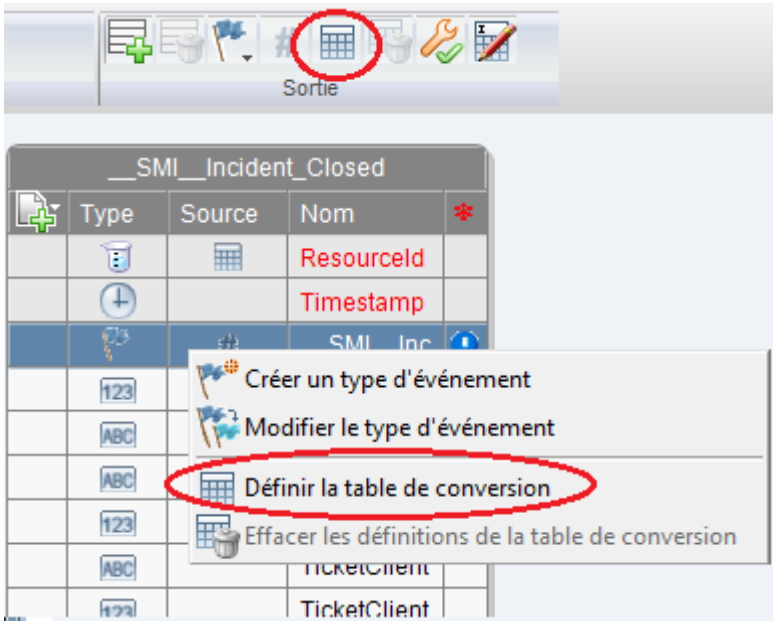


L'étape suivante consiste à configurer les éléments ResourceId, Horodatage et Type d'événement. Si les champs existent déjà (ils doivent être créés lors des étapes précédentes dans le cas contraire), ils peuvent être liés à ces champs, si nécessaire.

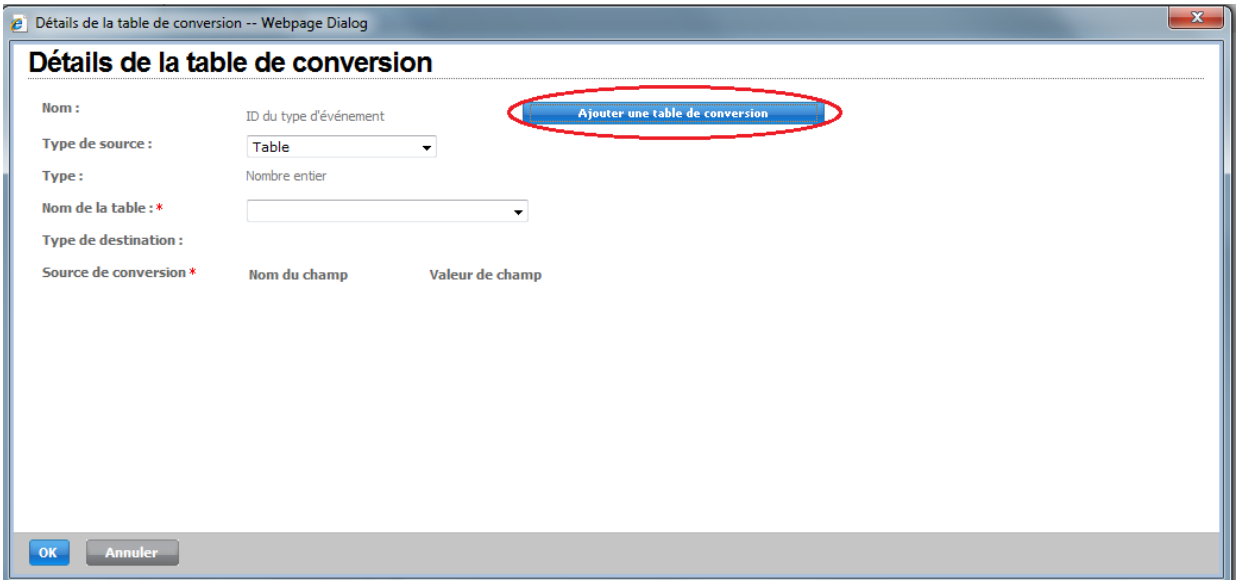
L'interface prend en charge un style d'association Glisser-déposer. Si l'élément associé ne demeure pas après sa définition, assurez-vous que le type de chacun correspond. (c.-à-d., Heure/chaîne/nombre entier, etc.).



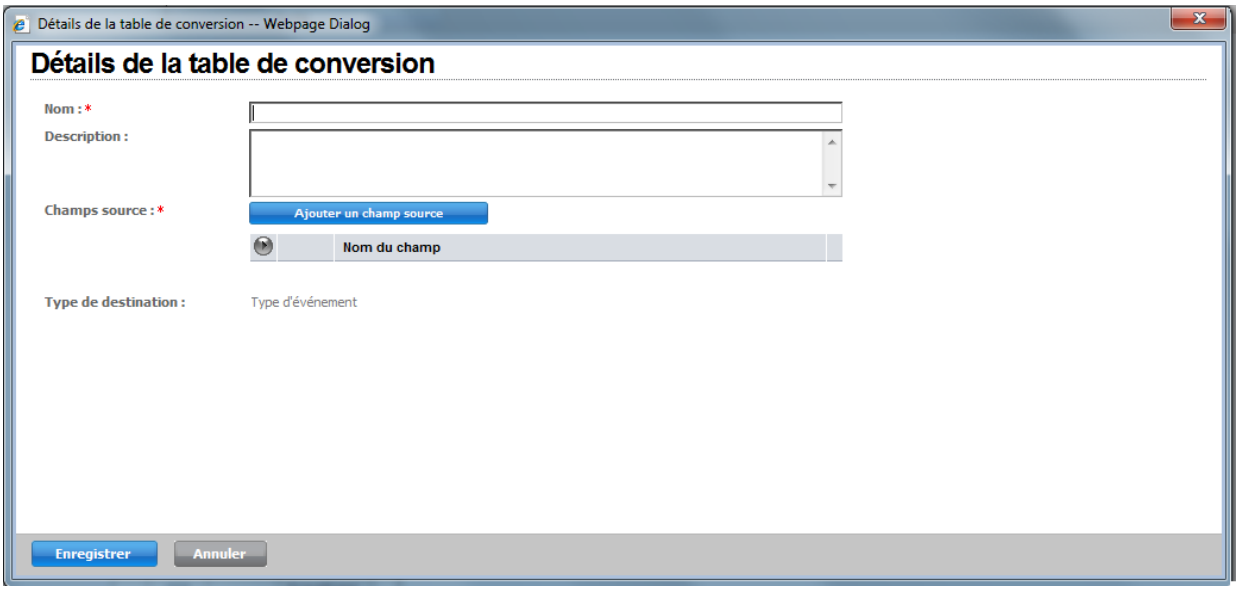
- L'élément ResourceId doit être mappé conformément aux exigences (identifiées pendant l'analyse) d'une valeur de chaîne dans l'entrée
- L'horodatage doit être défini sur la variable Heure qui définit la période où l'événement a eu lieu et doit être utilisé à des fins de calcul
- Le Type d'événement se définit par défaut sur une valeur constante conformément au Champ Nom de la source de données de la fenêtre précédente. Vous pouvez la remplacer si nécessaire. Cela serait nécessaire si plusieurs événements doivent être envoyés à partir de cet adaptateur, selon la valeur d'un champ spécifique (c.-à-d., vous voulez envoyer un autre événement selon la valeur d'un champ Entrée). Vous pouvez le faire en cliquant avec le bouton droit de la souris sur la ligne Type d'événement (ou cliquant sur le bouton de dessus comme indiqué dans l'image ci-dessous) et en choisissant Définir la table de conversion.



Après cela, une fenêtre contextuelle s'affiche.



Si la table de conversion pour ce champ de valeur existe déjà, vous pouvez la choisir ici, ou à défaut, définir une nouvelle table de conversion pour l'opération. Le bouton de la fenêtre de dessus peut être utilisé à cette fin.



Une fois que la table est créée, vous devez spécifier le champ Entrée mappé aux champs source indiqués ci-dessus.

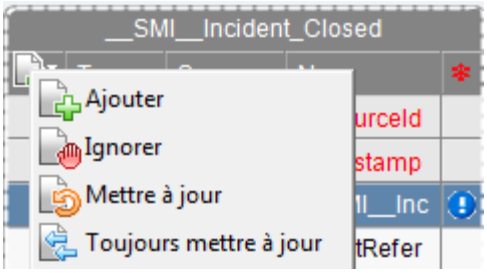
Si vous souhaitez spécifier une autre table de conversion pour les ressources de l'adaptateur, cela doit également s'effectuer à ce stade. Le processus de réalisation de l'opération est le même que celui décrit ci-dessus pour le Type d'événement.

Remarque : Par défaut, l'assistant de configuration d'adaptateur a affecté toutes les ressources à une table de conversion pré-existante appelée Default_Translation_Table sauf indication contraire. Ce peut être correct pour des implémentations simples mais, pour des implémentations plus complexes (et à des fins de séparation de données), CA recommande d'utiliser un tableau différent. Il est également obligatoire lorsque les champs source de la section de mappage d'adaptateur sont différents, ou contiennent plus d'une valeur.

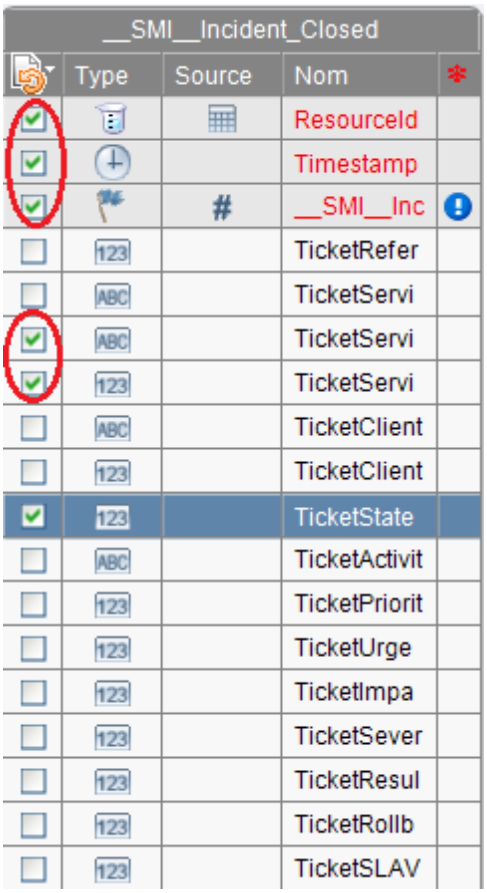
La dernière étape de l'étape Mappage consiste à configurer le paramètre OnDuplication de l'adaptateur. Ce paramètre décrit la mesure prise lorsqu'un deuxième événement, avec des valeurs correspondantes pour tous les champs clés, est reçu. Vous pouvez définir cette clé unique pour chaque sortie de l'adaptateur (pour plus d'informations sur cette opération, consultez la suite du document). Par défaut, cette valeur OnDuplication est définie sur Ajouter, par conséquent, seules les exigences à modifier sont nécessaires s'il faut prendre une autre mesure. Les valeurs disponibles sont les suivantes :

- *Ajouter* : le nouvel événement est seulement ajouté en tant que nouvel événement distinct
- *Ignorer* : le nouvel événement est ignoré (supprimé) par l'adaptateur
- *Mettre à jour* : le nouvel événement remplace l'événement précédemment chargé uniquement si les champs Valeur des événements sont différents
- *Toujours mettre à jour* : le nouvel événement remplace l'événement précédemment chargé

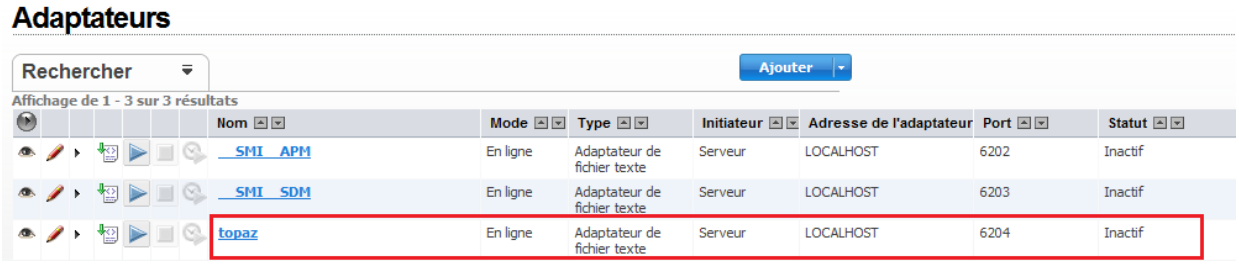
L'utilisation d'options autres que l'option Ajouter peut affecter les performances de l'adaptateur et doit soigneusement être prise en compte avant l'implémentation, particulièrement lorsque les volumes de données sont très importants.



Si la valeur est définie sur toute autre option différente d'Ajouter, la structure de sortie affiche une série de cases à cocher qui permettent de définir la clé unique. La clé comporte chacun des éléments avec une coche en regard. Le choix de la clé doit être décidé selon les besoins après une analyse attentive.



Une fois la configuration de la section de mappage terminée, cliquez le bouton Terminer dans le coin inférieur droit de la fenêtre. Il vous permet de retourner à la liste d'adaptateurs du système, et vous devez à cet instant pouvoir apercevoir l'adaptateur créé avec le statut Inactif.



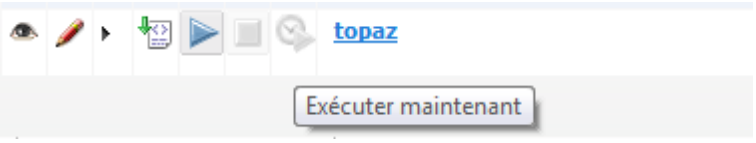
Déploiement d'adaptateur

Maintenant que l'adaptateur a été configuré, vous devez déployer l'adaptateur dans le serveur d'application. Cliquez sur le bouton Démarrer l'adaptateur pour permettre au service de déploiement de l'adaptateur de déployer l'adaptateur dans le système de fichiers. Cette opération inclut les actions suivantes :

- Créer les dossiers requis pour exécuter l'adaptateur
- Copier la configuration XML dans le dossier
- Créer un raccourci pour exécuter l'adaptateur

Une fois que cela a été fait, vos modifications sont appliquées sur le serveur.

Dès cet instant, il est maintenant possible d'exécuter l'adaptateur de l'interface utilisateur graphique, en cliquant sur le bouton Exécuter maintenant devenu actif. Cela permet au service de déploiement d'adaptateur d'exécuter l'adaptateur sur le serveur et de commencer la collecte de données.

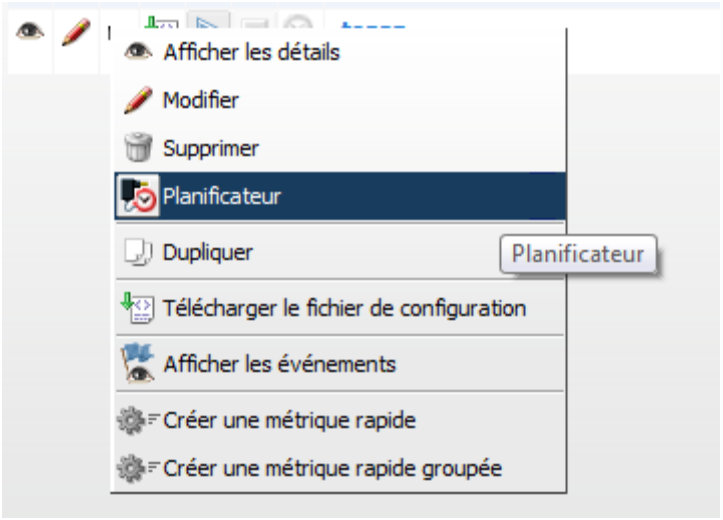


Une fois l'exécution de l'adaptateur terminée, vous devez ensuite pouvoir visualiser les Ressources et Evénements en attente sous la section de Conversions en attente du système.

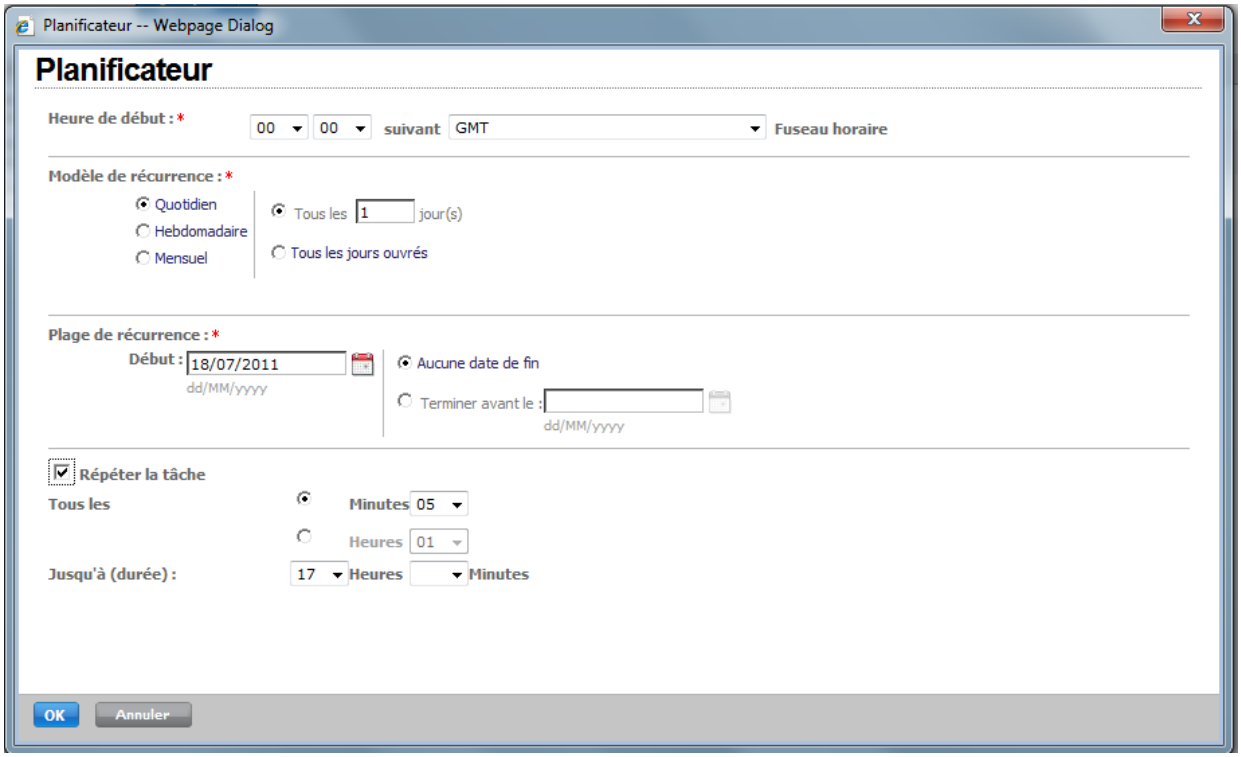
Vous pouvez alors convertir les Entrées en attente conformément à la configuration normale du système. Une fois la conversion terminée, l'adaptateur doit être exécuté de nouveau afin de charger les données brutes dans le système.

Planification d'adaptateur

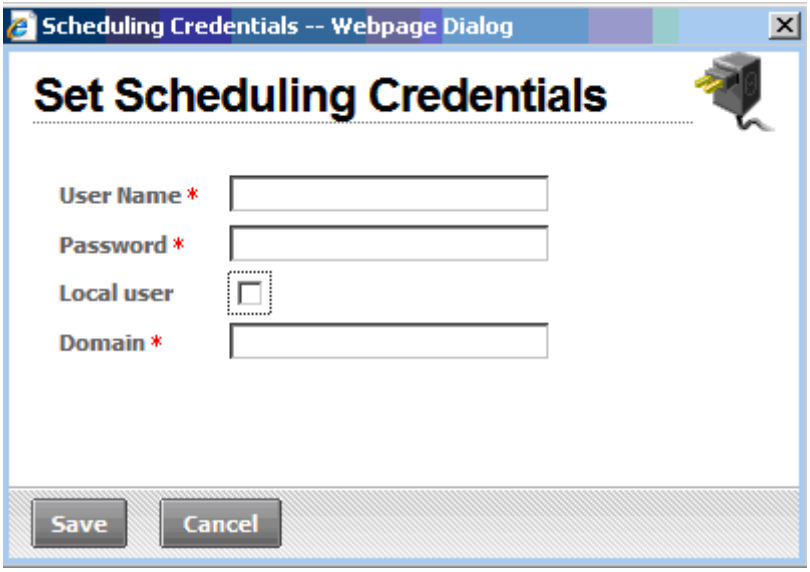
Outre l'exécution de l'adaptateur, il est également possible de définir une planification pour l'adaptateur depuis l'interface utilisateur graphique. Pour cela, l'adaptateur doit toutefois avoir le statut Arrêté. Une fois qu'il est arrêté, vous pouvez définir la planification en affichant le menu contextuel de l'adaptateur et en choisissant Planificateur.



Les options de planification sont les mêmes que celles fournies dans le Planificateur de tâches Windows. En arrière-plan, le service de déploiement de l'adaptateur crée cette planification en tant qu'élément du planificateur de tâches.



Après l'ajout d'une planification, et lorsque l'adaptateur est ensuite déployé, l'utilisateur est invité à saisir les informations d'identification de l'utilisateur du Compte sur le serveur qui accomplit la tâche. Elles doivent être saisies en tant que compte de service créé pour exécuter le système CA Business Service Insight (OblicoreSrv par défaut), ou un autre compte administratif si nécessaire.



Ce planificateur intégré est une fonctionnalité très utile puisqu'il permet à l'utilisateur de l'interface utilisateur graphique de contrôler la planification de l'adaptateur sans devoir accéder directement à l'ordinateur de bureau du Serveur d'application (sous réserve des autorisations d'utilisateur appropriées, bien entendu).

Etude de cas 21 : Exemple d'intégration de LDAP

Condition d'organisation

Utilisez les utilisateurs existants définis dans le serveur LDAP d'organisation. De plus, le portail d'organisation permet de se connecter et d'accéder à CA Business Service Insight à l'aide des fonctionnalités de connexion silencieuse de CA Business Service Insight pour les portails Single Sign On (SSO).

Définissez un script de conversion Visual Basic (VB) pour la création automatique des utilisateurs dans le système CA Business Service Insight (synchronisation LDAP), le script de conversion permet de se connecter au serveur LDAP d'organisation et d'en extraire la liste d'utilisateurs. Les méthodes de package d'outils CA Business Service Insight sont utilisées pour la création des utilisateurs, groupes et rôles.

Exemple de code VB de connexion au serveur LDAP

```
Option Explicit On
Imports System.DirectoryServices
    Public Function GetLDAPUsers(ByVal ldapServerName As String, ByVal pFindWhat
As String) As ArrayList
        Dim oSearcher As New DirectorySearcher
        Dim oResults As SearchResultCollection
        Dim oResult As SearchResult
        Dim RetArray As New ArrayList
        Dim mCount As Integer
        Dim mIdx As Integer
        Dim mLDAPRecord As String
        Dim ResultFields() As String = {"securityEquals", "cn"}
        Try
            With oSearcher
                .SearchRoot = New DirectoryEntry("LDAP://" & ldapServerName & _
          "/dc=lippogeneral,dc=com")
                .PropertiesToLoad.AddRange(ResultFields)
                .Filter = "cn=" & pFindWhat & "*"
                oResults = .FindAll()
            End With
            mCount = oResults.Count
            If mCount > 0 Then
                For Each oResult In oResults
                    mLDAPRecord =
oResult.GetDirectoryEntry().Properties("cn").Value & "  " &
oResult.GetDirectoryEntry().Properties("mail").Value
                    RetArray.Add(mLDAPRecord)
                Next
            End If
        Catch e As Exception
```

```
            MsgBox("Error is " & e.Message)
            Return RetArray
        End Try

        Return RetArray
    End Function

Sub CheckAddUser
Dim map
Set map = Tools.GetUserDetails("acme@Test")
'Vérifiez que l'utilisateur existe
'Mappage Tools.AddUserByMap
'Vérifiez les éléments dupliqués
map("UserName") = "acme2"
map("UserPassword") = "acme2"
map("UserPasswordExpirationInterval") = "50"
map("UserDescription") = "New description"
map("UserStatus") = "INACTIVE"
Tools.AddUserByMap map
Tools.Commit
End sub
```

Méthodes de script de conversion VB de CA Business Service Insight

- Méthodes d'organisation

  AddOrganization/IsOrganizationExists

- Méthodes de rôles

  IsRoleExists/SearchRoles

- Méthodes utilisateur

  AddUserByMap/GetUserName

  GetOrganizationName/IsUserExists

  GetUserDetails/SearchUsers

  GetUserFullName/UpdateUserByMap

- Méthodes de groupe d'utilisateurs

  AddUserGroupByMap/IsUserGroupExists

  DeleteUserGroup/SearchUserGroups

  GetUserGroupDetails/UpdateUserGroupByMap

Créez un code de connexion silencieuse et intégrez-le au portail d'organisation à utiliser pour la connexion au système CA Business Service Insight.

Exemple de code Gatway C# du système CA Business Service Insight (sera intégré dans le portail d'organisation)

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Web;
using System.Web.Security;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Security.Cryptography.X509Certificates;
using OblicoreAuthenticationWebService;

namespace Oblicore.SSO
{
      /// <summary>
      /// This sample page is a sample gateway to Oblicore Guarantee(tm)
application interface
      /// The page should be called prior navigating to any page at Oblicore
Guarantee website
      /// or any page using Web Services supplied by Oblicore
      /// The OblicoreGateway page should perform the following actions:
      ///         1) Call Oblicore Authentication Web service in order to
authenticate current user
      ///         2) Call SilentLogin.asp page at Oblicore website to login
silently at Oblicore website
      ///            and create user session context
      ///         3) Redirect to desired page
      /// </summary>
      public partial class _Default : System.Web.UI.Page
      {

            /// <summary>
            /// Oblicore user credentials
            /// </summary>
            struct UserCredentials
            {
                  public string UserName;
                  public string Organization;
            }

            private void Page_Load(object sender, System.EventArgs e)
            {
```

```
                if (Request["OGSESSIONID"]!=null)
                {
                      //We have been redirected back to this page from
SilentLogin.asp after authentication.
                      //Save OGSESSIONID in cookie for further use
                      HttpCookie SessionCookie = new
HttpCookie("OGSESSIONID",Request["OGSESSIONID"]);
                      Response.Cookies.Add(SessionCookie);
                      //Redirect to desired page
                      Response.Redirect("/");
                }
                else
                {
                      //First time we enter the page.
                      //Let's perform authentication.

                      string sAuthToken = string.Empty;

                      // Obtain OG user name and organizations from portal user
directory
                      UserCredentials ucOblicoreUser =
GetOblicoreUserCredentials();

                      //Initialize Oblicore Authentication WebServce
                      //Project should include Web Reference to the service
                      //Service is located on Oblicore Guarantee website at
/WebServiceN/DblicoreAuth.asmx
                      OblicoreAuth oAuthService = new OblicoreAuth();
                   //     oAuthService.ClientCertificates.Add(x509);
                      oAuthService.Url = "https://" + "localhost" +
"/WebServiceN/DblicoreAuth.asmx";
                      try
                      {
                            //Invoke authentication Web Service.
                            //The AuthenticateUser method returns encrupted
token, which should be passed to
                            //SilentLogin.asp page, located in root folder of
Oblicore Guarantee website
                            sAuthToken =
oAuthService.AuthenticateUser(ucOblicoreUser.UserName,ucOblicoreUser.Organization
);
                      }
                      catch (Exception ex)
                      {
                            //Proceed authentication error if any
                            Response.Write("The error has occurs during
Oblicore authentication: " + ex.Message);
                            Response.End() ;
                      }
```

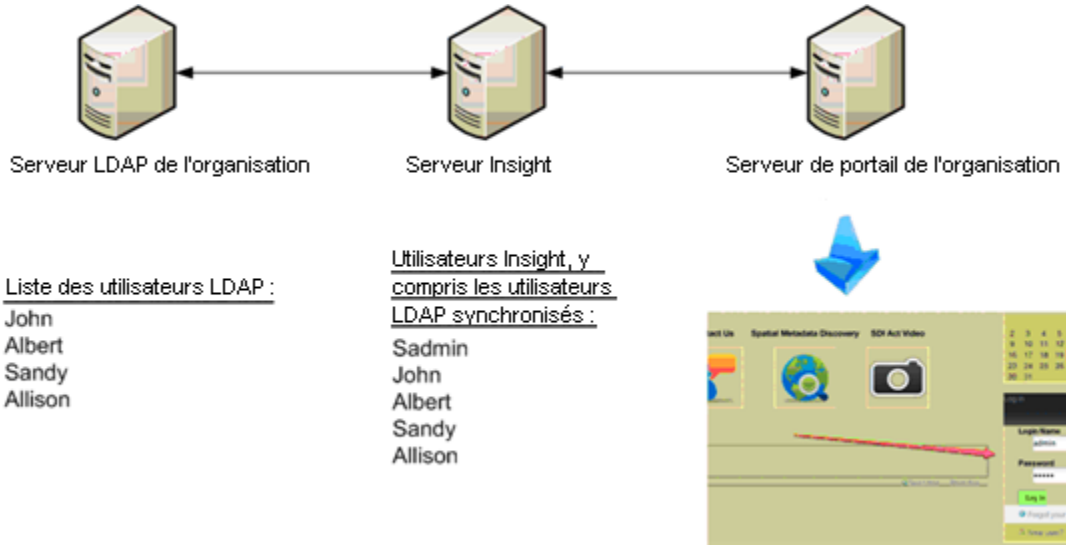
```
                        //Call SilentLogin.asp page along with passing it
authentication folder
                        //SilentLogin.asp page is located Oblicore Guarantee
website root folder
                        //After logging in, SilentLogin.asp page will redirect us
back to the current page along with passing OGSESSIONID parameter

//Response.Redirect(ConfigurationSettings.AppSettings["OGURL"].ToString() +
"/SilentLogin.asp?AuthToken="+Server.UrlEncode(sAuthToken)+"&DesiredPage="+GetCur
rentPageURL());
                        Response.Redirect("https://vit-
05/SilentLogin.asp?AuthToken=" + Server.UrlEncode(sAuthToken) +
"&DesiredPage=/Oblicore.asp"); // + GetCurrentPageURL());
                  }
            }


            /// <summary>
            /// Obtain Oblicore Guarantee user name and organization from portal
user directory
            /// The method is supposed to call ActiveDirectory or another
repository using portal API
            /// to obtain current user name and organization in terms of Oblicore
Guarantee
            /// </summary>
            /// <returns>Oblicore Guarantee user credentials struct</returns>
            private UserCredentials GetOblicoreUserCredentials()
            {
                  UserCredentials ucOblicoreUser = new UserCredentials();
                  //currently alwasy assume user is sadmin and organization is
Oblicore (default)
                  ucOblicoreUser.UserName = "sadmin";
                  ucOblicoreUser.Organization = "Oblicore";
                  return ucOblicoreUser;
            }



            /// <summary>
            /// Retrieves current page URL
            /// </summary>
            /// <returns>Full URL of current page</returns>
            private string GetCurrentPageURL()
            {
                  string s =
(Request.ServerVariables["HTTPS"]==null||Request.ServerVariables["HTTPS"].ToLower
()=="off")?"http://":"https://";
```

```
                  s += Request.ServerVariables["SERVER_NAME"] +
Request.ServerVariables["URL"];
                  if (Request.QueryString.ToString() != string.Empty)
                  {
                        s += "?"+Request.QueryString.ToString();
                  }
                  return s;
            }

            #region Web Form Designer generated code
            override protected void OnInit(EventArgs e)
            {
                  //
                  // CODEGEN: This call is required by the ASP.NET Web Form
Designer.
                  //
                  InitializeComponent();
                  base.OnInit(e);
            }

            /// <summary>
            /// Required method for Designer support - do not modify
            /// the contents of this method with the code editor.
            /// </summary>
            private void InitializeComponent()
            {
                  this.Load += new System.EventHandler(this.Page_Load);
            }
            #endregion
      }
}
```

- Organigramme

  L'organigramme suivant présente aux utilisateurs le processus de synchronisation et le flux d'accès au portail. Le script de conversion est configuré pour s'exécuter périodiquement. Le script maintient la liste d'utilisateurs LDAP jusqu'à ce jour et ajoute/supprime des utilisateurs selon le besoin.

  Les utilisateurs se connectent au portail d'organisation. Vous pouvez configurer le portail pour les rediriger au serveur CA Business Service Insight, ou afficher une liste d'autres applications disponibles. Le serveur CA Business Service Insight utilise les informations d'identification qui ont été fournies pendant la première connexion au portail.


Serveur LDAP de l'organisation
Serveur Insight
Serveur de portail de l'organisation
Liste des utilisateurs LDAP :
John
Albert
Sandy
Allison
Utilisateurs Insight, y compris les utilisateurs LDAP synchronisés :
Sadmin
John
Albert
Sandy
Allison

Etude de cas 22 : Génération de rapports à l'aide de PERL

L'exemple suivant démontre comment utiliser le script Perl pour se connecter au service Web de génération rapport CA Business Service Insight et transférer le paramètre XML des critères dans une enveloppe SOAP à l'aide d'un flux HTTP.

Remarque : Le code XML en cours de transfert dans l'enveloppe SOAP ne peut pas contenir les symboles < ou >, à la place du code HTML pour ces symboles (c.-à-d., <=> >=<)

```
#!/usr/bin/perl

#use strict;

use LWP::UserAgent;
use HTTP::Request::Common;
use XML::Simple;
use Data::Dumper;

my $userAgent = LWP::UserAgent > new(agent => 'Mozilla/5.0');

### Création d'un contexte de session Oblicore - L'ID de session Oblicore est
enregistré dans $scid ###

my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <CreateSessionContext xmlns=\"http://www.oblicore.com\">
      <userName>sadmin</userName>
      <organizationName>Oblicore</organizationName>
    </CreateSessionContext>
  </soap:Body>
</soap:Envelope>";

my $response = $userAgent > request(POST
'http://obonob/WebServiceN/DblicoreAuth.asmx',Content_type => 'text/xml', Content
=> $message);

print $response > error_as_HTML unless $response > is_success;

my $result = $response > as_string;

my $xmlstart      = index $result, "<?xml";
my $xmlend       = length $result;

my $xmltext       = substr $result, $xmlstart, ($xmlend - $xmlstart);
```

```
my $xml               = new XML::Simple;
my $data             = $xml > XMLin($xmltext);

my $scid = ($data > {'soap:Body'} > {'CreateSessionContextResponse'} >
{'CreateSessionContextResult'});

print "Session ID : ".$scid."\n";

 ### Essayer d'obtenir une liste de contrat d'Oblicore ###

my $criteria_xml =
"&lt;Criteria&gt;&lt;Name&gt;a*&lt;/Name&gt;&lt;Status&gt;EFFECTIVE&lt;/Status&gt
;&lt;/Criteria&gt;";

print  "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <GetContractsAdvanced xmlns=\"http://www.oblicore.com\">
      <sessionID>".$scid."</sessionID>
      <criteriaXML>".$criteria_xml."</criteriaXML>
    </GetContractsAdvanced>
  </soap:Body>
</soap:Envelope>";

my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <GetContractsAdvanced xmlns=\"http://www.oblicore.com\">
      <sessionID>".$scid."</sessionID>
      <criteriaXML>".$criteria_xml."</criteriaXML>
    </GetContractsAdvanced>
  </soap:Body>
</soap:Envelope>";

#my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
#<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
#  <soap:Body>
#    <GetContracts xmlns=\"http://www.oblicore.com\">
#      <sessionID>".$scid."</sessionID>
#    </GetContracts>
#  </soap:Body>
#</soap:Envelope>";
```

```
### is_well_formed($message)

print $message;

my $response = $userAgent > request(POST
'http://obonob/WebServices/Contracts.asmx', Content_Type => 'text/xml', Content
=> $message);

print $response > error_as_HTML unless $response > is_success;

my $result = $response > as_string;

print Dumper($result); # Output of contract list

### Fermer la session Oblicore ###

my $message = "<?xml version=\"1.0\" encoding=\"utf-8\"?>
<soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">
  <soap:Body>
    <ClearSessionContext xmlns=\"http://www.oblicore.com\">
      <sessionContextID>".$scid."</sessionContextID>
    </ClearSessionContext>
  </soap:Body>
</soap:Envelope>";

my $response = $userAgent > request(POST
'http://obonob/WebServices/OblicoreAuth.asmx', Content_Type => 'text/xml',
Content => $message);

print $response > error_as_HTML unless $response > is_success;
```

Annexe C : Spécifications de la configuration d'adaptateur

Ce chapitre traite des sujets suivants :

Structure globale (page 377)
Section General (page 378)
Section d'interface de CA Business Service Insight (page 385)
Section DataSourceInterface (page 388)
Section d'interface de SQL (page 392)
Section InputFormatCollection (page 397)
Section TranslationTableCollection (page 403)
Section TranslatorCollection (page 405)

Structure globale

La structure générale du fichier XML de configuration est la suivante :

```
< AdapterConfiguration>
    <General…>
    <OblicoreInterface…>
    <DataSourceInterface…>
    <InputFormatCollection…>
    <TranslatorCollection…>
    <TranslationTableCollection…>
</ AdapterConfiguration>
```

Chacun des noeuds est décrit dans les parties suivantes.

Section General

La section General comporte des attributs et sous-noeuds comme suit :

Structure XML :

```
<General
MajorVersion="2345"
MinorVersion="01245"
WorkingDirectoryName=" output"
DataSourceControlFileName="DatasourceControl.xml"
SendFileName="Send.txt"
SendControlFileName="SendControl.xml"
LogDebugMode="no"
ConsoleDebugMode="no"
RunOnce="yes"
RepeatUntilDry="yes"
RuntimeErrorLimit="1"
RetryRejectedEvents="yes"
RejectedEventsFileName="rejectedEvents.txt"
RejectedEventsUpperLimit="1000"
RegExUnlimitedSearch (V3.1 Patch)="no"
HoldRejectedEventsSpan="24"
GenerateStatistics="yes"
StatisticsFileName="MyStatistics.txt" >
KeepHistoricState="yes" >
DefaultTimeFormat="%Y/%m/%d-%H:%M:%S"
DefaultDecimalSymbol="."
DataSourceIdMaxLength="60"
DefaultDigitGroupingSymbol=","
SaveIllegalEvents ="no"
WriteEventErrorToLog ="yes"
IllegalEventsDirectoryName="xxxpath"
<DataSourceDifferenceFromUTC
        DefaultOffset="2"
        TimeFormat="%Y/%m/%d-%H:%M:%S">
        <DayLightSaving
                From="2001/04/09-12:00:00"
                To="2001/09/01-12:00:00"
                Shift="1"/>
</ DataSourceDifferenceFromUTC >
</General>
```

- MajorVersion : spécifie le numéro de version majeure.
- MinorVersion : spécifie le numéro de version mineure.
- WorkingDirectoryName : (facultatif)

  Spécifie le chemin d'accès par défaut des fichiers de sortie d'adaptateur (contrôle de source de données, conversion, envois).

Valeur par défaut : répertoire de sortie (dossier) du répertoire contenant le fichier de configuration.

- DataSourceControlFileName : (facultatif)

  Nom du fichier que l'adaptateur utilise pour suivre les derniers enregistrements de données récupérés à partir de la source de données.

  Valeur par défaut : DataSourcecontrol.xml (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- SendFileName : (facultatif)

  Nom du fichier qui conserve les événements avant leur envoi à CA Business Service Insight.

  Valeur par défaut : Send.txt (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- SendControlFileName : (facultatif)

  Nom du fichier que l'adaptateur utilise pour suivre le processus d'envoi.

  Valeur par défaut : SendControl.xml (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- DataSourceIdMaxLength :

  Longueur maximale du champ DataSourceId de la table t_raw_data. Si l'utilisateur insère une chaîne dépassant cette longueur, une erreur se produit dans l'adaptateur.

  Cette valeur doit être inférieure ou égale à la taille réelle de la table.

  Valeur par défaut : 60

- SaveIllegalEvents : s'il est sélectionné, cet attribut indique que les événements non autorisés doivent être écrits dans le fichier.

  Valeur par défaut : no

- WriteEventErrorToLog : permet de déterminer si les erreurs de données sont écrites dans la table T_Log.

  Valeurs autorisées : [yes/no]

  Valeur par défaut : yes

- IllegalEventsDirectoryName (pas de valeur par défaut) :

- SendFileName : (facultatif)

  Nom du fichier qui conserve les enregistrements avant leur envoi à CA Business Service Insight.

Valeur par défaut : Send.txt (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- §SendControlFileName : (facultatif)

  Nom du fichier que l'adaptateur utilise pour suivre le processus d'envoi.

  Valeur par défaut : SendControl.xml (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- LogDebugMode : (facultatif)

  Valeurs autorisées : [yes/no]

  Lorsque cet attribut est défini sur yes, ce qui suit est envoyé dans le journal : ligne d'origine à partir de la source de données, résultats de l'analyse, Evénement unifié CA Business Service Insight.

- ConsoleDebugMode : (facultatif)

  Valeurs autorisées : [yes/no]

  lorsque cet attribut est défini sur yes, les messages de débogage s'affichent sur la console.

  - **Indicateurs de lecture et de conversion des enregistrements** :
    - **.** : pour un enregistrement lu et converti en un événement de sortie.
    - **i** : pour un enregistrement ignoré par l'analyseur d'expression régulière.
    - **I** : pour un enregistrement lu et ignoré par les tables de conversion.
    - **R** : pour un enregistrement lu, mais rejeté par la table de conversion (ne peut être converti par les tables de conversion).
    - **X** : pour un enregistrement ayant posé problème lors de son analyse syntaxique. Il sera ignoré et perdu ou enregistré dans le répertoire des événements non autorisés.

  Remarque : Lorsque l'enregistrement lu passe par plus d'un convertisseur, l'indice d'enregistrement est mis entre parenthèses. Par exemple : [...] ou [RRI].

  - **Indicateurs de statut de l'adaptateur** :
    - **0** : l'adaptateur est actif et ne lit pas d'enregistrement à la dernière seconde.
    - **E** : statut Erreur.
    - **P** : statut Pause.
    - **S** : commande d'arrêt reçue de CA Business Service Insight.

- **B** : statut Bloqué, la table d'événements rejetés est saturée.
- **Indicateurs de tables de conversion** :
- **L** : attente des tables de conversion.
- **T** : table de conversion envoyée par/reçue de CA Business Service Insight.
- **t** : changements de la table de conversion envoyée par/reçue de CA Business Service Insight.
- **Indicateurs de connexion** :
- <Connecting CA Business Service Insight : tentative de connexion à CA Business Service Insight.
- <Connecting CA Business Service Insight> : connexion établie.
- <Connecting DataSource : tentative de connexion à la source de données.
- <Connecting DataSource> : connexion établie.

- RunOnce : (facultatif)

  Valeurs autorisées : [yes/no]

  lorsqu'il est défini sur no, l'adaptateur s'exécute continuellement.

  Lorsqu'il est défini sur yes, l'adaptateur de fichier texte s'exécute, lit des enregistrements et s'arrête automatiquement. L'adaptateur de fichier lit des fichiers entiers, attend quelques secondes, essaie de lire de nouveaux enregistrements (selon la durée de mise en veille). S'il n'existe pas d'enregistrements de ce type, il s'arrête.

  Un adaptateur SQL exécute chaque requête une seule fois. Si RepeatUntilDry est défini sur no, il s'arrête immédiatement. Si RepeatUntilDry est défini sur yes, il attend (selon la durée de mise en veille), essaie d'exécuter les requêtes à nouveau (selon la durée de mise en veille de la requête) et en l'absence de nouveaux enregistrements, il s'arrête.

- RepeatUntilDry : (facultatif)

  Valeurs autorisées : [yes/no] Valeur par défaut : yes

  Voir l'attribut RunOnce ci-dessus.

- RuntimeErrorsLimit : (facultatif)

  Détermine la limite des erreurs (par exemple, pour des erreurs des événements d'entrée) avant le blocage de l'adaptateur. Lorsque sa valeur est égale à 0, l'adaptateur n'est pas bloqué.

Valeur par défaut : 1 (ce qui signifie que l'adaptateur est bloqué après la première erreur.)

- RetryRejectedEvents : (facultatif)

  Valeurs autorisées : [yes/no] Valeur par défaut : yes

  Lorsque l'attribut est défini sur yes, les enregistrements qui doivent être convertis sont conservés dans le fichier d'événements rejetés, en attendant la conversion.

  Il est déconseillé de définir cet attribut sur no, car vous pourriez perdre des données importantes.

- RejectedEventsFileName : (facultatif)

  Nom du fichier que l'adaptateur utilise pour suivre les enregistrements de données récupérées de la source de données et qui attendent la conversion.

  Valeur par défaut : rejected.txt (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- RejectedEventsUpperLimit : (facultatif)

  Détermine la limite d'enregistrements rejetés conservés dans le fichier rejected.txt. Lorsque l'adaptateur atteint cette limite supérieure, il s'arrête et passe au statut bloqué jusqu'à ce que ces enregistrements soient convertis.

  Valeur par défaut : 1000

- RegExUnlimitedSearch : guide l'adaptateur dans la recherche complète conformément à l'expression régulière.

  Valeurs autorisées : [yes/no]

  Valeur par défaut : no

- HoldRejectedEventsSpan : (facultatif)

  Détermine le nombre d'heures pendant lesquelles il faut enregistrer le fichier d'événements rejetés. Si cet attribut n'est pas mentionné, l'adaptateur n'efface pas d'événement devant être géré avant le redémarrage de l'adaptateur.

  Il est déconseillé d'utiliser cet attribut, car vous pourriez perdre des données importantes.

- GenerateStatistics : (facultatif)

  Valeurs autorisées : [yes/no] Valeur par défaut : yes

  Lorsqu'il est défini sur yes, l'adaptateur crée un fichier de statistiques contenant des informations statistiques toutes les minutes.

- StatisticsFileName : (facultatif)

  Nom du fichier de statistiques.

  Valeur par défaut : statistics.txt (sauf indication contraire, c'est la valeur par défaut qui est utilisée.)

- KeepHistoricState : (facultatif)

  Valeurs autorisées : [yes/no] Valeur par défaut : no

  Lorsqu'il est défini sur yes, l'adaptateur enregistre tous les fichiers, sauf le fichier journal, dans un nouveau répertoire appelé Historic state jjmmaaaa-hhmmss, où jjmmaaaa et hhmmss représentent respectivement les formats de date et d'heure de création.

- DefaultTimeFormat : (facultatif)

  Format de date et d'heure par défaut. Si cet attribut est spécifié, il est utilisé comme format de date et d'heure à tout endroit où l'attribut TimeFormat est ignoré. S'il n'est pas spécifié, l'attribut TimeFormat dans les autres éléments est obligatoire.

- DefaultDecimalSymbol : (facultatif)

  Symbole décimal par défaut pour les champs réels.

  Valeur par défaut : (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- DefaultDigitGroupingSymbol : (facultatif)

  Symbole de groupement par défaut des chiffres pour le nombre entier et les champs réels.

  Valeur par défaut : (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- DataSourceDifferenceFromUTC : indique le décalage horaire entre l'UTC et le fuseau horaire de la source de données de l'ordinateur.
  - DefaultOffset : différence avec l'heure UTC lorsque la zone horaire n'est pas en heure d'été.
  - TimeFormat : spécifie le format selon lequel les détails des heures d'été (décrits dans la suite du texte) doivent être analysés.
  - DayLlightSaving : spécifie une période en heure d'été du fuseau horaire de la source de données. Cet élément est facultatif (ce qui signifie que s'il n'est pas sélectionné, il n'y a pas d'heures d'été indiquée) et il peut exister plus d'une fois. Lorsque certains éléments existent, ils doivent être ordonnés par heure et les périodes ne doivent pas empiéter les unes sur les autres.

- From : date de début de la période.
- To : date de fin de la période.
- Shift : période horaire ajoutée à DefaultOffset pendant la période en heure d'été.

Section d'interface de CA Business Service Insight

La section d'interface de CA Business Service Insight consiste en attributs spécifiant la connexion à CA Business Service Insight. Il existe deux modes : en ligne et hors ligne. En mode en ligne, l'adaptateur se connecte à CA Business Service Insight, récupère les tables de conversion et la commande de contrôle à partir de CA Business Service Insight et envoie les événements, les statuts et les requêtes de conversion à CA Business Service Insight. En mode hors ligne, l'adaptateur fonctionne avec un fichier de tables de conversion local et écrit les événements dans un fichier de sortie.

Structure XML du mode en ligne :

```
<OblicoreInterface
        Mode="online">
        <OnlineInterface
                Port="3006"
                ConnectionInitiator="server"
                Address="localhost"
                SecurityLevel="high"
                SecurityKey="12345678901234567890123456789012"
                UseAcknowledgeProtocol="yes"
                PacketSize="50"
                PacketDeadline="60"
                PacketResendTimeout="60"
                WindowSize="10"
      />
</OblicoreInterface>
```

- Port : numéro de port TCP/IP que l'adaptateur utilise pour communiquer avec le serveur CA Business Service Insight.
- ConnectionInitiator : (facultatif)

  Valeurs autorisées : server/adaptater. Valeur par défaut : server.

  Définit l'initiateur de connexion, l'adaptateur ou l'écouteur d'adaptateur sur CA Business Service Insight.

  L'initiateur joue le rôle d'esclave ou de client, tandis que l'autre est le maître ou joue le rôle de serveur.
- Adresse : adresse réseau du serveur ; obligatoire lorsque l'initiateur est l'adaptateur.
- SecurityLevel : définit le niveau d'authentification entre l'adaptateur et le serveur CA Business Service Insight (poignée de main). Options de niveaux :
  - none : aucune authentification n'est effectuée.

- high : l'authentification est effectuée. La clé SecurityKey doit être fournie.

- SecurityKey : chaîne de 32 chiffres. Ce doit être la même chaîne que celle définie sur l'adaptateur dans la base de données CA Business Service Insight.

  Le flux du processus poignée de main est le suivant :

  - Le serveur CA Business Service Insight se connecte à l'adaptateur.
  - Une chaîne au hasard est envoyée de l'adaptateur au serveur CA Business Service Insight.
  - Le serveur CA Business Service Insight chiffre la chaîne avec une clé préconfigurée et renvoie le résultat à l'adaptateur.
  - L'adaptateur chiffre la même chaîne aléatoire qui a été envoyée au serveur CA Business Service Insight à l'aide de la chaîne SecurityKey et compare les résultats.
  - Le serveur CA Business Service Insight constitue une chaîne et envoie la chaîne à l'adaptateur.
  - L'adaptateur chiffre la chaîne et la renvoie au serveur CA Business Service Insight.
  - Le serveur CA Business Service Insight chiffre la même chaîne et compare les résultats à ceux reçus de l'adaptateur.
  - La connexion est établie.
  - Si une défaillance est détectée dans l'une des phases du flux, la connexion n'est pas établie.

- UseAcknowledgeProtocol : (facultatif)

  Valeurs autorisées : [yes/no] Valeur par défaut : yes

  Lorsque cet attribut est défini sur yes, l'adaptateur utilise le protocole d'acquittement. Lorsqu'il est défini sur non, l'adaptateur envoie les messages/paquets et n'attend pas le message d'acquittement de CA Business Service Insight.

  Il est déconseillé de définir cet attribut sur no, car vous pourriez perdre des données importantes.

- PacketSize : (facultatif)

  Valeurs autorisées : de 1 à 1000

  Valeur par défaut : 50

  Nombre maximal d'événements dans un paquet.

- PacketDeadline : (facultatif)

Valeurs autorisées : de 1 à 3600

Valeur par défaut : 60

Nombre de secondes écoulées avant l'envoi d'un paquet qui n'est pas complet.

- PacketResendTimeout : (facultatif)

  Valeurs autorisées : de 1 à 3600

  Valeur par défaut : 60

  Délai d'attente (en secondes) d'un message d'acquittement avant renvoi du paquet. Cet attribut est applicable uniquement lorsque l'attribut UseAcknowledgeProtocol est défini sur yes.

- WindowSize (facultatif) :

  Valeurs autorisées : de 1 à 100

  Valeur par défaut : 10

  Nombre de paquets dans la fenêtre. Cet attribut est applicable uniquement lorsque l'attribut UseAcknowledgeProtocol est défini sur yes.

Structure XML pour le mode hors ligne :

```
<OblicoreInterface
        Mode="offline">
        <OfflineInterface
                OutputFileName="outputEvents.txt"
                OutputFileType="tsv"
                OutputTimeFormat="%Y/%m/%d %H:%M:%S"
                OutputDecimalSymbol="."
        />
</OblicoreInterface>
```

- OutputFileName : (facultatif)

  Nom du fichier dans lequel l'adaptateur écrit les événements de sortie.

  Valeur par défaut : AdapterOutput.txt (sauf indication contraire, c'est la valeur par défaut qui est utilisée).

- OutputFileType : (facultatif)

  Valeurs autorisées : csv/tsv/xml

  Définit le format d'événement.

- OutputTimeFormat : définit le format des champs de date. Vous pouvez ignorer cet attribut si l'attribut DefaultTimeFormat, dans la section General, a été défini.

- OutputDecimalSymbol : (facultatif)

Définit le symbole décimal pour des champs réels.

Valeur par défaut : . (point)

Section DataSourceInterface

La section DataSourceInterface comporte des attributs spécifiant la connexion et le type de connexion entre l'adaptateur et la source de données (Outil de mesure, CRM, Journal système, etc.) et est divisé en deux types principaux : l'Interface Fichier et l'interface SQL.

Interface Fichier

Vous pouvez utiliser l'adaptateur de fichier pour récupérer des données des fichiers journaux, des rapports planifiés ou tout autre fichier texte, et DataSourceInterface définit des règles d'analyse syntaxique des informations depuis la source de données de fichier et leur extraction dans des champs.

La section DataSourceInterface définit également la manière dont l'adaptateur gère le fichier source (s'il supprime le fichier d'origine au cas où il avait été créé uniquement pour l'adaptateur, ou s'il laisse les données seules au cas où elles serviraient pour d'autres utilisations, et ainsi de suite).

Structure XML :

```
<DataSourceInterface WorkFileName="MyWorkingFile.txt" >
   <Files>
        <File
            IsActive="yes"
                InputFormat="events"
                Path="D:\adapters\sample_data\"
                NamePattern="adapterXXX*.log"
                DeleteFileAfterProcessing="no"
                Delimiters=","
                IgnoreRedundantDelimiters ="no"
                TitleExists="no"
                SleepTime="10">
```

```
        </File>
    </Files>
</DataSourceInterface>
```

- WorkFileName (facultatif) : Lorsque DeleteFileAfterProcessing est défini sur no - le fichier est copié sous ce nom, lorsqu'il est défini sur yes - le fichier est renommé sous ce nom. S'il n'est pas spécifié, la valeur par défaut sera considérée (WorkFile.txt).
  - Files : collection d'éléments de fichiers (vous pouvez définir plus d'un fichier dans un adaptateur).
  - File : spécifie les attributs du fichier.
    - IsActive (facultatif) [yes/no]. Définit si ce fichier est actif (lorsqu'il est défini sur no, ce fichier ne sera pas lu).
    - InputFormat : format de fichier associé à ce fichier. L'adaptateur utilise l'InputFormat pour extraire les données de la source de données.
    - Path : chemin d'accès menant vers l'emplacement des fichiers de données sources.
    - NamePattern : spécifie le nom de fichier de source de données. Peut utiliser un caractère générique si plus d'un fichier utilise le même format d'entrée.
    - DeleteFileAfterProcessing [yes|no] : la façon dont l'Adaptateur gère le fichier source. Période à laquelle le fichier a été créé uniquement pour l'adaptateur ; vous pouvez le supprimer après le traitement en le définissant sur oui. Le fichier est renommé, traité et supprimé. Lorsqu'il est défini sur non, le fichier est copié et le traitement a lieu dans le fichier copié. Si les nouveaux enregistrements sont ajoutés à la fin de ce fichier, l'adaptateur copie ces nouveaux enregistrements dans le fichier de travail dans le cycle suivant. Si les nouveaux enregistrements ne sont pas ajoutés au fichier, l'adaptateur cherche le premier fichier avec le même modèle et le nom plus long (dans l'ordre lexicographique) que le fichier actuel. Si l'Adaptateur trouve un tel fichier, il entreprend le travail avec ce fichier. L'adaptateur ne retourne pas au fichier précédent même si les nouveaux enregistrements sont ajoutés à ce fichier. Utilisez non lorsque vous devez conserver l'intégrité du fichier source.
    - InitialFileName : le premier nom de fichier à partir duquel l'adaptateur cherchera un fichier avec un modèle donné. Utilisez cet attribut lorsque le NamePattern contient des caractères génériques et vous ne voulez pas que l'adaptateur lise d'anciens fichiers.

- Délimiteurs (facultatif) : Un ou plusieurs caractères servant de délimiteurs, d'après lesquels des lignes de données sont analysées dans des champs ; si aucune valeur n'est spécifiée, la valeur par défaut est "\t".
- IgnoreRedundantDelimiters (facultatif) : [yes/no]. Lorsqu'ils sont définis sur yes, les délimiteurs consécutifs seront traités comme un seul, sinon un champ vide sera créé entre les délimiteurs.
- RegExForParser (facultatif) : Expression régulière à utiliser afin d'extraire des champs de la source de données remplaçant les délimiteurs spécifiés plus tôt. Par exemple :

```
<File
    ….
    RegExForParser="^(.{8}) (.{6}) (?:[ ])*([0-9]+) "
/>
```

  Consultez la documentation sur les expressions régulières pour plus de détails.

- TitleExists (facultatif) [yes/no] : spécifie si la première ligne non vide dans le fichier de source de données est une ligne de titre. Le titre peut être utilisé par l'adaptateur lors de l'analyse des données.
- SleepTime : spécifie l'intervalle de données de récupération en secondes) c.-à-d. - nombre de secondes de pause entre l'extraction de données de l'Adaptateur du fichier de données source.
- LogicLineDefinition (facultatif) :

```
<File
    ….
    <LogicLineDefinition
                    FirstLine="Job server:.*"
                    NumberOfLines="5"
    />
/>
```

  Au cas où l'ensemble de données est créé à partir du nombre de lignes et non par une ligne, les attributs suivants définissent le point de départ d'extraction, le terminal et le numéro de lignes compromettant les données :

  - AllFile (facultatif) [yes/no] : lorsque défini sur yes, tout le fichier est considéré comme un seul enregistrement, une ligne logique.
  - FirstLine (facultatif) : expression régulière qui définit la première ligne de la ligne logique. Vous pouvez le spécifier avec ou sans les attributs LastLine et/ou NumberOfLines.
  - LastLine (facultatif) : expression régulière qui définit la dernière ligne de la ligne logique. Vous pouvez le spécifier avec ou sans les attributs FirstLine et/ou NumberOfLines.

- NumberOfLines (facultatif) : numéro de lignes dans une ligne logique. Vous pouvez le spécifier avec ou sans les attributs FirstLine et/ou LastLine.
- MatchCase (facultatif) [yes/no] : définit si l'expression régulière correspondante doit respecter la casse.

Section d'interface de SQL

Vous pouvez utiliser l'adaptateur SQL pour récupérer des données des bases de données en utilisant une instruction SQL.

L'interface de SQL définit la connexion à la base de données et les requêtes utilisées pour récupérer les données :

Structure XML :

```
        < DataSourceInterface >
                <ConnectionString ConnectionTimeout="60" QueryTimeout="30">
                        <![CDATA[ Driver={Microsoft Access Driver
(*.mDataBase)};DataBaseq=d:\Oblicore\database1.mdatabase; ]]>
                        </ConnectionString>
                <QueryCollection>
                <Query QueryName="cases" InputFormat="cases" SleepTime="3600">
                        <SelectStatement AutoCompleteQuery="yes">
                        select
dateclosed,callid,dateopened,companyname,priority,closedmn,responsemn
                        from calls where dateclosed is not NULL
                        </SelectStatement>
                <QueryKeyFields>
                        <KeyField Name="dateclosed" Sort="asc"/>
                        <KeyField Name="callid" Sort="desc"/>
                        <SelectInitialValues>
                                Select min(dateclosed) , 'min date' from calls
                        </SelectInitialValues>
                </QueryKeyFields>
        </Query>
        <Query QueryName="contracts" InputFormat="contracts" SleepTime="3600">
        <ConnectionString>
                <Segment Type="text"
                Text=" Driver={Microsoft Excel Driver (*.xls)}; DriverId=790;
DataBaseq="/>
                <Segment Type="File">
                <File Path="d:\Oblicore " NamePattern="Availabilty_*.XLS>
                </Segment>
                <Segment Type="text" Text=";"/>
        </ConnectionString>
<SelectStatement AutoCompleteQuery="yes">….</SelectStatement>
<QueryKeyFields>…..</QueryKeyFields>
</Query>
</QueryCollection>
</DataSourceInterface>
```

- ConnectionString : chaîne de Connexion pour accéder à la base de données source.

Vous pouvez définir le ConnectionString dans l'élément DataSourceInterface et/ou dans les éléments Requête. La définition de ConnectionString dans l'élément de DataSourceInterface est la valeur par défaut et il est utilisé uniquement dans une requête sans une définition de ConnectionString.

Vous pouvez définir la chaîne de connexion dans une chaîne ou par segments. Lorsque l'élément ConnectionString ne contient pas d'éléments Segment, la chaîne de connexion est prise dans le texte de l'élément ConnectionString. Si elle contient au moins un élément de segment, la chaîne de connexion est concaténée à partir de cet élément.

Il existe deux types de segment. Le premier est le texte et contient le texte concaténé à la chaîne de connexion telle quelle. Le deuxième type est le fichier et contient un nom de fichier avec ou sans caractères génériques. Le segment de fichier ne peut s'afficher qu'une fois et contient d'autres attributs qui définissent quoi faire avec le fichier lu.

- ConnectionTimeout (facultatif) : nombre entier positif qui indique, en secondes, le temps à attendre avant que la connexion ne s'ouvre. La valeur 0 indique de patienter jusqu'à l'établissement de la connexion. La valeur par défaut est de 15 secondes.

Remarque : Certains fournisseurs ne prennent pas en charge cette fonctionnalité.

- QueryTimeout (facultatif) : nombre entier positif qui indique, en secondes, le temps d'attente avant l'exécution de la requête. La valeur 0 indique de patienter jusqu'à la fin de l'exécution de l'opération. La valeur par défaut est de 30 secondes.

Remarque : Certains fournisseurs ne prennent pas en charge cette fonctionnalité.

- Segment : spécifie les attributs du segment.
- Type (facultatif) (texte, fichier) : type de segment
- Text : texte du segment.
- File : spécifie les attributs du fichier
- Path : chemin d'accès menant vers l'emplacement des fichiers de données sources.
- NamePattern : spécifie le nom de fichier de source de données. Il peut contenir des caractères génériques.
- InitialFileName (facultatif) : indique à l'utilisateur à partir de quel fichier il doit commencer la recherche et le modèle à rechercher.
- UsePath (facultatif) [yes, no] : lorsque défini sur yes, le chemin d'accès au fichier est concaténé à la chaîne de connexion.

- UseFileName (facultatif) [yes/no] : lorsque défini sur yes, le nom de fichier est concaténé à la chaîne de connexion (nécessaire pour les fichiers Excel).
- UpdateSchemaFile (facultatif) [yes/no] : lorsque défini sur yes, l'adaptateur met à jour le fichier schema.ini avec le nom de fichier actuel.

Remarque : Utilisez cet attribut uniquement lorsque vous voulez que l'adaptateur change le fichier schema.ini (base de données pour des fichiers texte).

- QueryCollection : collecte de requêtes (plusieurs requêtes peuvent être définies dans un adaptateur).
  - PreliminaryCheck : (facultatif [yes/no]). L'adaptateur vérifie les requêtes lorsqu'il se connecte à la base de données. Si cet attribut est défini sur no, l'adaptateur contrôle les requêtes en les exécutant sans aucun changement, l'adaptateur s'exécute sur les enregistrements de retour plus tard et ne les lit pas à nouveau. S'il est défini sur yes, l'adaptateur remplace chaque chaîne where de l'énoncé select avec "where 1=2 and" et uniquement alors il exécute les requêtes. Utilisez cette option si vous voulez vérifier toutes les requêtes avant que l'adaptateur commence à s'exécuter sur les enregistrements et lorsque cet ajout réduit dramatiquement le temps de requête. Remarque : Certains fournisseurs n'optimisent pas le processus de requête et lorsque la requête est autorisée, l'exécution prend le même temps que celui nécessaire sans l'ajout.
  - ExternalCommand (facultatif) : la commande externe qui est exécutée devant l'adaptateur commence un nouveau cycle de lecture.
  - Command : nom du fichier de commandes qui doit être exécuté.
  - SleepTime : nombre entier positif qui indique, en secondes, le temps à attendre avant d'exécuter la commande à nouveau.
- Query : spécifie les attributs de la requête.
  - QueryName : nom de la requête.
  - IsActive (facultatif) [yes/no] : lorsque défini sur no, l'adaptateur ne lit pas les enregistrements de ce fichier.
  - InputFormat : format de fichier associé à cette requête. L'adaptateur utilise InputFormat pour extraire les données de l'enregistrement source.
  - SleepTime : durée en secondes au cours de laquelle l'adaptateur se met en veille en attendant l'arrivée de nouvelles données.

- Critical (facultatif) [yes/no] : Si défini sur yes, l'adaptateur s'arrête immédiatement lorsqu'il trouve une erreur dans la requête. Utilisez l'option non lorsqu'il s'agit d'une erreur connue qui est résolue sans changer le fichier de configuration.
- TransactionMode (facultatif) [implicit/explicit] : Si défini sur explicit, l'adaptateur initie une transaction de base de données avant la requête. Cette option résout plusieurs problèmes dans des requêtes immenses et compliquées.
- RollbackSegementName (facultatif) : définit quel segment d'annulation l'adaptateur utilise. Dans le cas contraire, la base de données choisit le segment d'annulation.

- SelectStatement : contient l'instruction à exécuter sur la base de données source.
  - AutoCompleteQuery (facultatif) [yes/no] : S'il est défini sur yes, l'adaptateur automatiquement concatène un énoncé where à la requête spécifiée comme suit :
  - crée un énoncé where qui apportera des valeurs uniquement nouvelles d'après les champs clé.
  - Tri des résultats en fonction des champs clé.
- QueryKeyFields : définit les champs pour commencer la récupération suivante de données :
  - KeyField :
  - Name : nom du champ, provenant des champs de la requête.
  - Sort (facultatif) [asc/desc] : Méthode de tri de données (croissante/décroissante).
  - Function (facultatif) : Utilisez cet attribut si une fonction spéciale doit s'appliquer à ce champ afin de combiner la valeur de champ et la fonction (:fieldname). Par exemple, vous pouvez utiliser la fonction de date d'Oracle avec un nom de champ Ts-Function="to_date(':ts','jj/mm/aaa')"
  - ValueLeftWrapper (facultatif) : Utilisez cet attribut pour concaténer des caractères avant la valeur du champ. La valeur par défaut des champs Chaîne et Date est "'" (apostrophe).
  - ValueRightWrapper (facultatif) : Utilisez cet attribut pour concaténer des caractères après la valeur du champ. La valeur par défaut des champs Chaîne et Date est "'" (apostrophe).

- NameLeftWrapper (facultatif) : Utilisez cet attribut pour concaténer des caractères avant le nom du champ. La valeur par défaut est une chaîne vide.
- NameRightWrapper (facultatif) : Utilisez cet attribut pour concaténer des caractères après le nom du champ. La valeur par défaut est une chaîne vide.
- IsPrimaryKey (facultatif) [yes/no] : Lorsqu'elle est définie sur no, cette clé n'est pas utilisée par l'adaptateur dans l'énoncé "where" automatique de la requête.
- DefaultInitialValue (facultatif) : Si la requête SelectInitialValues ne renvoie pas d'enregistrements, l'adaptateur prend la valeur par défaut qui se trouve dans cet attribut. S'il existe plusieurs champs clé, tous les champs clé doivent avoir cet attribut.
- SelectInitialValues : énoncé "select" qui fournit les valeurs initiales au premier énoncé "where" (lorsque le fichier de contrôle est vide).

Remarque : L'ordre des valeurs doit être identique à celui des éléments de champ de ce QueryKeyFields et renvoyer un résultat pour chaque champ.

Section InputFormatCollection

Cette section spécifie la structure de données récupérées à partir d'une source de données, la manière dont une ligne de données doit être découpée en champs et quels sont les formats et les types de champ. Le filtrage des données initiales et les manipulations de données peuvent être effectués dans cette section en utilisant les champs InputFormatSwitch et Compound respectivement.

Le flux de travaux général de cette section est le suivant :

- La ligne de données est comparée à une ou plusieurs spécifications InputFormats.
- Les données sont réparties dans des champs en fonction de la spécification InputFormat correspondante.
- Des valeurs sont assignées aux champs Compound en combinant et en scindant les champs de données.
- Les données traitées sont vérifiées par rapport aux conditions TranslatorSwitch.
- Les données traitées sont soit envoyées au convertisseur correspondant, soit ignorées.

Le noeud InputFormatCollection peut contenir un ou plusieurs noeuds InputFormat.

Structure XML :

```
<InputFormatCollection>
        <InputFormat InputFormatName="MyInputFormat">
        <InputFormatFields>
                <InputFormatField Name="sid_id" Type="string"/>
                <InputFormatField Name="content" Type="string"/>
                <InputFormatField Name="date" Type="time"
               TimeFormat="%d/%m/%Y %H:%M:%S"/>
                <InputFormatField Name="server" Type="string"
               Source="compound">
                <Compound>
                        <Segment SourceField="content"
                    RegularExpression=".*Job server: ([^\n]+).*" />
                </Compound>
                </InputFormatField>
        </InputFormatFields>
<TranslatorSwitch DefaultTranslator="GeoTranslator">
        <TranslatorCase  TranslatorName="NonGeoTranslator" Break="yes">
                <Condition SourceField="routing_info" Operator="EQ"
            Value="cnano"/>
```

```
        </TranslatorCase>
</TranslatorSwitch>
</InputFormat>
</InputFormatCollection>
```

- InputFormat :
  - InputFormatName : nom attribué à ce format et utilisé par la section DataSourceInterface.
- RequiredFields (facultatif) : nombre minimum de champs censé se trouver dans une ligne de données. Une ligne contenant moins de champs est ignorée et une erreur est consignée.
- InputFormatFields : peut contenir un ou plusieurs noeuds de champs selon le nombre de champs d'entrée dans les sources de données.
  - InputFormatField : spécifie un champ de données de la ligne de données d'origine ou un champ composé.

    ```
    <InputFormatField Name="timestamp" Type="time"
    TimeFormat="%d/%m/%Y %H:%M:%S"/>
    ```
  - Name : nom attribué à ce champ et utilisé par d'autres éléments.
  - Type : type de données de champ, à savoir string/integer/real/time
  - Source (facultatif) : la valeur par défaut est event, avec les valeurs possibles suivantes :
  - event : valeur du champ de l'événement provenant de la source de données, les valeurs de champs suivent le même ordre que celui de la source de données.
  - compound : le champ est composé. Sa valeur lui est attribuée une fois que d'autres valeurs ou constantes de champs ont été manipulées d'une manière ou d'une autre.
  - title : la valeur du champ provient du nom de champ de titre. Le champ de référence doit déjà être défini.
  - filename : la valeur du champ provient du nom de fichier de la source de données, uniquement pour des adaptateurs de fichier texte.
  - constant : la valeur du champ est constant et elle provient de la propriété ConstantValue qui doit s'afficher à sa suite.
  - FieldTitleName : lorsque la source est title, spécifie le titre de champ à utiliser. Le champ source doit être défini à l'avance.

- ConstantValue : quand la source est constant, spécifie la valeur à laquelle elle doit correspondre. Dans le cas d'un champ horaire, la valeur constant est formatée selon le format d'heure TimeFormat, Now ou NowUtc, Now correspondant à l'heure actuelle dans l'environnement linguistique de la source de données et NowUtc correspondant à l'heure en temps universel.
- TimeFormat : format de date et d'heure pour lequel le champ est analysé. Les codes de caractère suivants peuvent être utilisés pour les formats de date et d'heure :

Symbole	Description
%d	Jour (1 - 31)
%a	Nom abrégé du jour de la semaine (dim., lun., etc.)
%A	Nom complet du jour de la semaine (dimanche, lundi, etc.)
%m	Mois (1 - 12)
%b	Nom abrégé du mois (jan., fév., etc.)
%B	Nom complet du mois (janvier, février, etc.)
%y	Année (sur 2 chiffres)
%Y	Année (sur 4 chiffres)
%H	Heure au format 24 h (00 - 23)
%I	Heure au format 12 h (01 - 12)
%p	Indicateur AM/PM pour le format 12 h
%M	Minutes (0 - 59)
%S	Secondes (0 - 59)
%D	Date complète sans séparateurs (jjmmaaaa)
%T	Heure complète sans secondes et sans sépateurs (hhmm)
%n	Nombre de secondes depuis 1970
%E	Date et heure au format hexadécimal
%%	Symbole de pourcentage

- DecimalSymbol (facultatif) : symbole décimal des champs. Si ce n'est pas spécifié, le champ DefaultDecimalSymbol de la section General est utilisé.

- DigitGroupingSymbol (facultatif) : symbole de groupement de chiffre par défaut pour les champs Integer et Real. Si ce n'est pas spécifié, le champ DefaultDigitGroupingSymbol de la section General est utilisé.

- Compound : obligatoire lorsque source=compound. Spécifie les manipulations de champ qui doivent être rassemblées dans un champ composé.
  - Segment : spécifie une manipulation de champ à ajouter au composé créé. Seul l'attribut SourceField est requis.
  - SourceField : champ utilisé en tant que source. Le champ de référence doit déjà être défini.
  - RegularExpression : expression régulière de manipulation.
  - MatchCase (facultatif) [yes/no] : définit si l'expression régulière respecte la casse.
  - SelectionStart : position de démarrage de l'extraction de texte, part de 0.
  - SelectionLength : taille de l'extraction de texte.
  - Prefix : chaîne devant recevoir un préfixe pour obtenir le résultat de la manipulation.
  - Suffix : chaîne devant recevoir un suffixe pour obtenir le résultat de la manipulation.
  - XpathExpression : expression xpath de manipulation.
- InputFormatSwitch : utilisé pour spécifier des critères de format, lorsque les lignes de données n'ont pas toutes le même format.

  Remarque : Lorsque vous utilisez InputFormatSwitch, l'ordre des noeuds InputFormat est important ; un champ de référence InputFormat doit déjà être défini.

  DefaultInputFormat : spécifie le nom InputFormat vers lequel il doit être dirigé, au cas où aucun critère ne soit satisfait.
- InputFormatCase : spécifie un critère à tester sur des lignes de données pour déterminer vers quel InputFormat il doit être dirigé.
- InputFormatName : le champ InputFormat à utiliser lorsque le critère correspond.
- LogicOperator : (facultatif) [and/or].
- and : toutes les conditions doivent être satisfaites. (valeur par défaut)
- or : au moins une condition doit être satisfaite

- Condition : condition à tester sur une ligne de données pour déterminer son format.

  SourceField : champ à tester.

  Operator : type de test des options suivantes :

  - **EQ** : égal à
  - **NE** : pas égal à
  - **GT** : supérieur à
  - **LT** : inférieur à
  - **GE** : supérieur ou égal à
  - **LE** : inférieur ou égal à
  - **MATCH** : une expression régulière doit concorder
  - **UNMATCH :** une expression régulière ne doit pas concorder

  ValueType (facultatif) [constant/field/previousValue] :

  - constant : le contenu de l'attribut Value est constant indépendamment des données source
  - field : le contenu de l'attribut Value correspond au nom de champ du même enregistrement.
  - previousValue : le contenu de l'attribut Value correspond au nom de champ de l'enregistrement précédent dans la même requête avec le même format d'entrée.

  Value : valeur à faire concorder ou une expression régulière.

  MatchCase (facultatif) [yes/no] : définit si le test respecte la casse. Si défini sur yes, les deux valeurs sont converties en minuscules avant le test.

- TranslatorSwitch : détermine quel convertisseur doit être utilisé pour convertir la ligne de données en un événement unifié CA Business Service Insight.

  - DefaultTranslator : convertisseur à utiliser lorsqu'aucun critère n'est satisfait. Si la valeur est Ignore, aucun convertisseur n'est utilisé et la ligne est ignorée.
  - TranslatorCase : spécifie les critères à tester sur les données traitées afin de déterminer vers quel convertisseur il doit être dirigé.

    Break [yes|no]

    yes : (valeur par défaut) si les critères sont satisfaits, ne vérifiez pas les critères suivants.

no : dans tous les cas, après avoir évalué les critères et fait fonctionner le convertisseur s'il concorde, passez au critère suivant.

LogicOperator : (facultatif) [and/or].

- and : toutes les conditions doivent être satisfaites. (valeur par défaut)
- or : au moins une condition doit être satisfaite.

TranslatorName : le convertisseur vers lequel se diriger si les conditions sont remplies.

Condition : la condition à tester sur des données traitées afin de déterminer quel convertisseur pertinent doit être utilisé. Il s'agit de la même que la Condition de InputFormatSwitch.

Section TranslationTableCollection

La section contient des tables de mappage qui font correspondre les valeurs de source de données aux champs d'événement de CA Business Service Insight.

Structure XML :

```
<TranslationTableCollection
        LoadingMode="remote"
        TranslationTablesFileName="Translations.xml">
        <TranslationTable
                Name="ResourcesTranslateTable"
                DestinationType="resource">
                <TranslationField>nodeName</TranslationField>
        </TranslationTable>
        <TranslationTable
                Name="EventTypesTranslateTable"
                DestinationType="event_type">
                <TranslationField>eventType</TranslationField>
        </TranslationTable>
        <TranslationTable
                Name="valueUpDownTranslateTable"
                DestinationType="value"
                ValueType="string">
                <TranslationField>eventType</TranslationField>
        </TranslationTable>
</TranslationTableCollection>
```

- TranslationTablesFileName : (facultatif) spécifiez le nom de fichier dans lequel les tables sont stockées localement ; si non spécifié, la valeur par défaut est choisie ("Translation.XML").
- LoadingMode : (facultatif) [standalone, remote].

  Remarque : La valeur par défaut pour l'interface en ligne est Remote, et Standalone pour l'interface hors ligne. Méthode de chargement spécifiée des tables de conversion :

- standalone : l'adaptateur charge les tables de conversion localement. Aucune connexion avec le serveur de CA Business Service Insight quant à la conversion. Les changements au sein des tables de conversion seront stockés uniquement dans le fichier local.
- remote : l'adaptateur envoie une demande afin de charger toutes les tables du serveur de CA Business Service Insight. Les changements apportés dans les tables de conversion seront aussi stockés localement.
- LoadTimeout : (facultatif) lorsque le mode de chargement est distant, vous pouvez spécifier ici une temporisation en secondes.
  - TranslationTable : lie la valeur d'événement à la table de mappage.

- Name : nom qui sera utilisé et référencé par le convertisseur. Le nom légal commence par une lettre ou un soulignement et contient des lettres, des chiffres et des soulignements.
- DestinationType : [resource, event_type, contract_party, service, time_zone, value].
- ValueType : (facultatif) [integer, real, string] type de la valeur renvoyée par la table. Les valeurs de chaîne et réelles sont légales uniquement pour la table avec la mention DestinationType="value"
- TranslationField : le nom de champ à traduire, le nom du champ est pris des champs de format d'entrée. Vous pouvez avoir jusqu'à 5 champs.

Section TranslatorCollection

La section TranslatorCollection décrit la méthode de conversion de l'enregistrement de la source de données analysée et manipulée, extrait dans la section précédente vers un événement CA Business Service Insight.

Lorsque le mode d'interface est "en ligne", l'événement CA Business Service Insight a une structure unifiée qui contient les champs suivants :

- Timestamp : moment de l'occurrence d'événement.
- ResourceId : ID de ressource de CA Business Service Insight associée à l'événement (la ressource qui a été mesurée, etc.).
- EventTypeId : type d'événement CA Business Service Insight associé à l'événement, décrit le type de l'événement (type de mesure sur la ressource, type d'action de ticket, etc.) .
- DataSourceId : (facultatif) le nom de la source de données d'entrée (nom de fichier/nom de table, etc.).
- Value : la valeur de l'événement (résultat de mesure, numéro de ticket, etc.). Ce champ peut apparaître plusieurs fois.

Lorsque le mode d'interface est "hors ligne", le nombre de champs ou leur nom ne sont soumis à aucune restriction.

Structure XML :

```
<TranslatorCollection>
   <Translator TranslatorName="events" OnDuplication = "ignore">
        <TranslatorFields>
                <TranslatorField Name="ResourceId" SourceType="table"
                    SourceName="ResourcesTranslateTable" Iskey="yes"/>
                <TranslatorField Name="EventTypeId"      SourceType="constant"
                    ConstantValue="1002" Iskey="yes"/>
                <TranslatorField Name="Timestamp"        SourceType="field"
                SourceName="timestamp" Iskey="yes"/>
                <TranslatorField Name="Value"    SourceType="table"
                SourceName="valueUpDownTranslateTable" Iskey="yes"/>
                < TranslatorField Name="Value"  SourceType ="field"
                SourceName ="nodeName" Iskey="yes"/>
                < TranslatorField Name="Value" SourceType ="constant"
                Type="integer" ConstantValue="1000" Iskey="yes"/>
                < TranslatorField Name="Value"  SourceType ="field"
                SourceName ="timestamp" TimeShift="-3600"
                TimeShiftFieldName="createDate"  Iskey="yes"/>
                < TranslatorField Name="Value" SourceType ="lookup"
                SourceName ="ServiceTable" LookupValue="word"
                Iskey="yes"/>
```

```
      </TranslatorFields>
   </Translator>
</TranslatorCollection>
```

- Translator : décrit la méthode de conversion de l'ensemble de champs qu'il reçoit dans un événement de sortie.
  - TranslatorName : nom utilisé par InputFormat pour envoyer des définitions de champ à ce convertisseur.
  - OnDuplication : membre qui détient la valeur "ignore","add", "update" ou "updateAlways" pour déterminer comment traiter un événement de duplication. Consultez la section Singularité d'événement.
  - TranslatorFields : contient la liste des éléments TranslatorField, chacun d'eux renfermant les attributs suivants :
  - Name : Nom de champ. Dans l'interface en ligne, ce doit être Timestamp, ResourceId, EventTypeId, Value ou DataSourceId.
  - SourceType :

    field : la valeur du champ provient du champ de format d'entrée. L'attribut SourceName contient le nom de champ.

    table : la valeur du champ provient de la table de conversion. L'attribut SourceName contient le nom de la table.

    lookup : la valeur du champ provient de la table de conversion. L'attribut SourceName contient le nom de la table. La valeur à convertir est prise à partir de l'attribut LookupValue et non du format d'entrée.

    constant : la valeur du champ est constante et se trouve dans l'attribut ConstantValue.
  - SourceName : contient le nom de champ pour le nom de la table de conversion.
  - Type : [integer/real/string/time] requis uniquement lorsque le type du champ n'est pas prédéfini (par nom de champ ou par SourceType). Dans l'interface en ligne, requis uniquement pour le champ Value lorsque SourceType=constant. Dans l'interface hors ligne, requis pour chaque champ lorsque SourceType=constant.
  - IsKey : représente la clé unique de l'événement. Cette clé est assemblée à partir de plusieurs champs qui ont été marqués TranslatorFields?IsKey = "yes".

    Consultez la section Singularité d'événement.
  - LookupValue : Contient la valeur de l'élément de recherche lorsque SourceType="lookup".

- ConstantValue : Contient la valeur constant lorsque SourceType=constant. Dans le cas d'un champ horaire, la valeur constant est formatée selon le format d'heure TimeFormat, Now ou NowUtc, Now correspondant à l'heure actuelle dans l'environnement linguistique de la source de données et NowUtc correspondant à l'heure en temps universel.
- TimeFormat : contient le TimeFormat, requis uniquement pour des champs lorsque SourceType=constant et Type=time.
- TimeShift : définit le nombre d'heures de différence en secondes, uniquement pour les champs d'heure.
- TimeShiftFieldName : (facultatif) contient le nom de champ du format d'entrée qui indique le nombre d'heures de différence en secondes. TimeShift et TimeShiftFieldName peuvent être ensemble.

Annexe D : Définition de formules de logique applicative (Expert en logiques applicatives)

Ce chapitre traite des sujets suivants :

Choses à éviter lors de la création de formules de logique applicative (page 409)
Métriques groupées et efficience des ressources (page 410)

Choses à éviter lors de la création de formules de logique applicative

Lors de la création de formules de logique applicative, veillez à garder à l'esprit les points suivants :

- N'affectez jamais une valeur nulle à une variable globale. L'attribution d'une valeur nulle peut entraîner une défaillance du moteur de calcul lors du calcul de la métrique. Si une valeur non initialisée est requise, utilisez plutôt Empty.
- Evitez d'utiliser des objets de carte et de vecteur dans les métriques groupées. Si vous devez utiliser de tels objets, veillez à ce qu'ils soient aussi petits que possible. Le calcul des métriques groupées contenant des vecteurs ou des cartes de grande taille prend beaucoup de temps.

Métriques groupées et efficience des ressources

Qu'est-ce qu'une métrique groupée ?

La métrique groupée permet de définir une métrique à utiliser pour chaque membre d'un groupe de ressources et ce, afin d'appliquer la même définition et la même logique à un ensemble d'éléments. Un groupement peut être défini statiquement sur un ensemble prédéfini de ressources ou dynamiquement sur les membres du groupe de ressources, tandis que le groupe peut être modifié dans le temps et inclure ou exclure des membres.

Vous pouvez exclure une ressource ou un groupe de ressources au groupe dans le temps, l'inclure voire même l'exclure, puis l'inclure de nouveau au groupe, maintes et maintes fois, dans la même période de calcul (jour, mois, année, etc.).

Que se passe t-il au niveau de la logique applicative lorsqu'un élément de groupe est supprimé du groupe de base de métrique groupée ?

La méthode OnPeriodEnd et la fonction Result sont déclenchées pour l'élément de groupe. Si cela se produit au milieu de la période de calcul, le résultat est uniquement consigné dans la base de données lorsque la période de calcul d'origine s'achève (p. ex. fin du mois, fin de l'année).

Que se passe t-il au niveau de la logique applicative lorsqu'un élément de groupe est ajouté au groupe de base de métrique groupée ?

Les variables globales sont initiées, les méthodes OnLoad, OnRegistration et OnPeriodStart sont déclenchées pour l'élément de groupe.

Que se passe t-il au niveau de la logique applicative lorsqu'un élément de groupe est ajouté au groupe de base de métrique groupée après avoir été supprimé du groupe au cours de la même période de calcul ?

Le résultat défini pour la période pendant laquelle l'élément de groupe faisait partie du groupe est remplacé par le nouveau résultat. En d'autres termes, le résultat à la fin de la période de calcul se réfère uniquement à la dernière période de la période calculée, lorsque l'élément de groupe fait partie du groupe.

Quel est l'impact de l'attribut de mise en vigueur d'une ressource sur la logique applicative ?

Lorsqu'une ressource est expirée, aucune donnée brute n'est collectée pour cette dernière.

Quel est l'impact de l'attribut de mise en vigueur d'une ressource sur le groupement ?

Le fait de changer une ressource pour qu'elle expire a le même impact sur le groupement que l'exclusion de la ressource du groupe. Le groupement a le même comportement que ce soit pour l'efficience de la ressource ou l'appartenance au groupe.

Comment implémenter des exceptions sur une ressource ? L'utilisation de l'efficience de ressource est-elle la bonne méthode ?

Il existe quelques études de cas pour lesquelles une période d'exception doit être définie sur une ressource spécifique, par exemple, un serveur peut être en maintenance et doit donc être exclu des calculs pendant la période de maintenance.

Dans la mesure où la logique applicative ignore les événements de données brutes d'une ressource qui est expirée, vous pouvez éventuellement choisir d'implémenter des exceptions sur une ressource à l'aide du mécanisme d'efficience. Cela peut être une solution dans certains cas. Toutefois, si la ressource fait partie d'une métrique groupée et que la ressource est en vigueur, puis expire durant la même période de calcul, seule la dernière période pendant laquelle la ressource a été en vigueur est prise en compte dans le résultat, comme indiqué au-dessus. Dans ce cas, il est recommandé d'utiliser la fonctionnalité d'attributs personnalisés. Vous pouvez gérer un attribut supplémentaire pour la ressource indiquant le statut de la ressource. La formule de logique applicative interroge alors le statut de la ressource au moment opportun dans le script.

Glossaire

Adaptateur

L'interface entre CA Business Service Insight et les sources de données tierces. Les adaptateurs convertissent les données issues de ces sources de données dans un format utilisable dans les calculs de niveau de service fournis aux contractants.

Agent

Un objet qui représente une métrique dans une unité d'heure donnée.

Alerte

Notifications aux utilisateurs des événements survenant dans le système. Les alertes permettent aux utilisateurs de mettre en oeuvre des actions correctives de manière proactive pour éviter tout écart par rapport aux contrats, d'encourir des pénalités, etc.

Allocation des ressources

Allouer une ressource à un service dirige le flux des événements de ressource vers ce service. Une ressource peut être allouée à un service, un contractant, un type ou un groupe.

Annotations d'événement

Informations supplémentaires sur un événement spécifique. Les annotations d'événement sont définies automatiquement ou manuellement (dans les rapports).

Assistant de création de rapports

Interface utilisateur graphique utilisée pour définir les paramètres des rapports.

Assistant de modification de métriques

Interface conviviale développée pour permettre aux utilisateurs de modifier facilement les métriques dès leur premier ajout au contrat à partir d'un modèle de niveau de service.

Auteur du contrat

Utilisateur en charge de la création d'un certain contrat.

Catalogue des services

Ensemble de toutes les informations relatives aux services, domaines et unités, à la bibliothèque de modèles et aux modèles de niveau de service dans CA Business Service Insight.

Catégorie de domaine

Aspect spécifique d'un niveau de service conformément au contrat. Imaginons par exemple que vous disposez d'un domaine de service appelé Assistance. Vous pouvez alors disposer de la catégorie de domaine Temps de réponse de l'assistance, qui décrit un aspect particulier du service d'assistance. L'administrateur système du prestataire de service définit généralement les catégories de domaine. CA Business Service Insight fournit également des catégories de domaine prédéfinies.

Commentaire manuel sur les causes premières

Commentaire, défini manuellement par la personne qui visualise le rapport, pour expliquer un résultat de niveau de service d'une métrique spécifique ou pour y ajouter des informations supplémentaires. Les commentaires manuels sur les causes premières peuvent être partagés par toutes les personnes appartenant à la même métrique qui consultent le rapport.

Commentaire sur les causes premières

Commentaire défini dans la logique applicative pour expliquer les résultats du niveau de service. Les commentaires sur les causes premières sont associés à des métriques.

Consommation

Type de métrique qui calcule la consommation. Ce type de métrique est principalement utilisé pour le module financier. Grâce à ce type de métrique, il est possible d'afficher la consommation et les prix séparément dans les rapports. Vous pouvez également calculer la consommation dans la métrique d'élément de prix.

Contractant

Client ou fournisseur avec lequel le contrat a été signé. Un contractant peut également représenter une entité interne au sein d'une entreprise plus grande.

Contrat archivé

Contrat dont les données ont été classées dans les archives. Les contrats archivés ne sont pas inclus dans les calculs et ne peuvent pas être affichés dans les rapports.

Contrat de sous-traitance

Contrat de sous-traitance. Contrat avec un prestataire de services externe pour qu'il assure la prestation de services de l'entreprise.

Conversions

Conversion des données assemblées par les adaptateurs des sources de données vers les entités définies dans CA Business Service Insight.

Date de saisie de la ressource

Date à laquelle la ressource a été entrée dans CA Business Service Insight. A ne pas confondre avec la date d'entrée en vigueur de la ressource.

Date d'entrée en vigueur de la ressource

Date à partir de laquelle CA Business Service Insight peut utiliser la ressource. La date d'entrée en vigueur de la ressource est définie à l'aide de la version de la ressource qui comprend la ressource. La ressource sera reconnue par une version de ressource dont la date d'entrée en vigueur est antérieure à la version actuelle de la ressource.

Dispositif d'alerte

Dispositifs du réseau qui reçoivent les messages d'alerte par courriel, affichage contextuel ou SMS ou via un programme tiers.

Dossier de rapports

Conteneur utilisé pour regrouper des rapports associés entre eux d'une manière ou d'une autre.

Elément de prix

Métrique qui calcule les prix. Les prix peuvent reposer sur la consommation ou être configurés en tant que prix fixes.

Enoncé d'objectif

Description logique d'une métrique contenant les paramètres qui définissent la métrique. Les énoncés d'objectif s'affichent dans l'interface utilisateur graphique de CA Business Service Insight pour garantir une capture efficace de l'essence de la métrique.

Ensemble de modifications

Ensemble des modifications apportées au mappage des ressources du prestataire de service.

Entrée de conversion

Représente une définition des valeurs sources et cibles utilisées par la table de conversion.

Evénement de données

Evénements générés par les adaptateurs CA Business Service Insight

Evénement de données brutes

Evénement généré par des données brutes dans CA Business Service Insight.

Evénement de métrique

Evénement généré par une métrique dans CA Business Service Insight.

Evénements reçus

Liste des événements reçus d'autres métriques présentées dans la fenêtre Portée de la logique applicative en cliquant sur l'onglet Evénements reçus.

Exception

Période de temps non prise en compte dans le calcul des niveaux de service. Par exemple, les arrêts imprévus.

Granularité

La granularité détermine les unités de temps supplémentaires dont l'auteur de la métrique souhaite obtenir le résultat pour exclure la période de suivi de la métrique.

Groupe de contractants

Groupe de contractants (clients) défini de manière logique. Faire référence à un groupe plutôt qu'à un contractant individuel rationalise la création des contrats. Un contractant peut appartenir à plusieurs groupes de contractants.

Groupe de rapports

Rapport regroupant plusieurs rapports réguliers, placés côte à côte.

Groupe de rapports

Rapport qui comprend plusieurs rapports, côte à côte, sur une page.

Groupe de ressources

Ensemble de ressources regroupées en une unité logique. Un groupe de ressources peut contenir une ou plusieurs ressources individuelles ou un ou plusieurs groupes de ressources.

KPI

Indicateur clé de performance. Mesure significative utilisée seule ou en combinaison avec d'autres indicateurs clés de performance pour surveiller dans quelle mesure une activité ou un service parvient à ses objectifs quantifiables.

KQI

Indicateur clé de qualité. Mesure significative utilisée seule ou en combinaison avec d'autres indicateurs clés de performance pour surveiller dans quelle mesure une activité ou un service parvient à ses objectifs de qualité.

Livret

Fichier RTF pouvant inclure les données du contrat et les rapports connexes dans un format pratique à l'aide de modèles prédéfinis et configurables.

Métrique

Combinaison de plusieurs paramètres définissant le niveau de service cible d'un certain service à un moment donné. Chaque métrique est associée à un seul domaine de service. Les champs et attributs de la métrique comprennent la catégorie de domaine, la formule de niveau de service, le service, la période d'application, l'objectif de niveau de service et la période de suivi.

Métrique connexe

Métrique référencée comme un objectif dans une relation.

Métrique de consommation

Métriques de consommation qui permettent d'afficher la consommation et les prix séparément dans un rapport.

Métrique dérivée

Métrique créée à partir d'un modèle de contrat ou d'un modèle de niveau de service.

Métrique groupée

Type de métrique qui s'applique à plusieurs ressources ou groupes de ressources.

Métrique informationnelle

Métrique qui exécute un calcul informatif à utiliser à des fins de reporting uniquement.

Métrique intermédiaire

Métrique pouvant générer des événements dans le but de calculer des événements. Il est impossible de calculer les métriques intermédiaires.

Modèle de contrat

Contrat prédéfini utilisé pour créer un nouveau contrat.

Modèle de logique applicative

Formules de logique applicative prédéfinies utilisables pour définir de nouvelles formules de logique applicative.

Modèle de niveau de service

Ensemble prédéfini de services et de métriques dont l'objet est de faciliter la création et la modification des contrats en fonction des processus commerciaux standard.

Modules de logique applicative

Bibliothèque de fonctions de scripts utilisables dans la logique applicative.

Moteur de statut actuel

Processus autonome qui calcule les métriques de statut actuel. Le nombre d'instances qu'il est possible d'exécuter sur un ou plusieurs ordinateurs n'est pas limité.

OLA

Accord sur les niveaux opérationnels. Un contrat OLA est un SLA dans lequel le fournisseur correspond à l'organisation interne et les clients constituent les entités commerciales internes.

Options régionales

Détails spécifiques à la région dans laquelle se trouve le contractant. Les paramètres incluent la langue, la devise, le décalage horaire par rapport au fuseau horaire GMT, l'heure d'été et le format de la date.

Package

Ensemble de modèles de niveau de service et de modèles réunis pour être installés et décompressés sur une autre instance de CA Business Service Insight.

Paramètre

Valeur définissable en dehors de la logique applicative pour affecter la formule de logique applicative. Les paramètres sont utilisés par la logique applicative pour déterminer le résultat du niveau de service. Un paramètre peut être de type texte, liste, chiffre, date ou table. Entre autres exemples de paramètres figurent les seuils et les noms de ressource.

Paramètre de contrat

Paramètre d'un contrat régulier, d'un modèle de contrat et d'un modèle de niveau de service que les métriques utilisent dans la logique applicative.

Paramètres de métrique

Paramètres d'une métrique.

Pénalité

Compensation due au contractant si l'objectif de niveau de service n'est pas satisfait dans la période de suivi définie. Les pénalités peuvent être activées ou désactivées dans CA Business Service Insight.

Période de suivi

Période d'application pendant laquelle CA Business Service Insight mesure le niveau de service fourni par rapport à l'objectif de niveau de service stipulé dans le contrat. La mesure prise pendant cette période détermine l'existence d'une déviation par rapport à l'objectif défini (par la logique applicative) et le risque de pénalités ou de primes. La période de suivi détermine également la méthode de génération des rapports d'activités.

Piste d'audit

Enregistrement chronologique des activités du système et de l'utilisateur CA Business Service Insight conservées dans le système. L'audit peut être contrôlé ultérieurement pour identifier les actions des utilisateurs sur le système ou les processus mis en oeuvre sur le système.

Piste d'audit du contrat

Journal de toutes les activités d'un contrat.

Plan du tableau de bord

Présentation du tableau de bord. Agencement des widgets dans le tableau de bord.

Profil d'alerte

Ensemble de paramètres définissant les conditions de déclenchement d'une alerte, les destinataires de cette alerte et son mode de transmission.

Proportion

Type d'unité.

Rapport au format libre

Un rapport créé par un utilisateur définit l'instruction SQL d'après la base de données de CA Business Service Insight ou d'autres bases de données externes.

Rapport composé

Plusieurs rapports réguliers qui s'affichent sous forme de plusieurs séries sur un seul graphique.

Rapport simple

Support permettant d'analyser les résultats calculés par CA Business Service Insight à partir d'un ensemble important de critères.

Relation

Entité décrivant une connexion d'un certain type établi entre deux métriques et ayant les mêmes propriétés.

Ressource

Elément unique de l'infrastructure complète du prestataire de services. Entre autres ressources figurent les serveurs, commutateurs, concentrateurs, routeurs, services d'assistance et autres éléments mesurables. Plusieurs ressources peuvent être définies comme appartenant à un groupe de ressources, ce dernier étant alors considéré comme une autre ressource.

Rôle

Ensemble de responsabilités, activités et autorisations. Le rôle de l'utilisateur définit le mode d'affichage dans CA Business Service Insight. Chaque utilisateur CA Business Service Insight reçoit un ou plusieurs rôles qui déterminent les actions qu'il peut exécuter dans CA Business Service Insight. Les actions qu'un rôle ne permet pas d'effectuer sont supprimées de l'interface utilisateur de CA Business Service Insight lorsque l'utilisateur accède à l'application.

Script de conversion

Un script de conversion peut faciliter la maintenance des conversions et prévenir les erreurs.

SLO

Objectif de niveau de service. Utilisé pour mesurer les contrats.

Table de conversion

Mécanisme utilisé pour convertir des valeurs, c'est-à-dire les données issues des données brutes en données définies dans le système.

Tableau de bord

Interface utilisateur prévue pour les dirigeants qui organise et présente les informations en temps réel tout en permettant la création de rapports facile à lire.

Type de métrique

Décrit le motif du calcul d'une certaine métrique et définit la liste des champs et le comportement en termes de champs obligatoires et par défaut qu'une métrique d'un type donné doit posséder.

Type de ressource

Catégorie de ressources intégrée. Une ressource doit être allouée à au moins un type de ressource.

Types d'événement

Un événement est une mesure appliquée à une ressource par un outil tiers, puis convertie par l'adaptateur en un format compatible avec CA Business Service Insight. Les types d'événements déterminent comment ces événements sont définis et formatés dans CA Business Service Insight.

Unité

Catalogue des unités mesurables. Exemples : pourcentage et devise.

Unité de mesure

Unité de mesure de toutes les métriques définies pour une catégorie de domaine.